Universidade do Minho
Escola de Engenharia

Carlos Eduardo Rodrigues Teixeira Salgado

An OMG Model-based Approach for Aligning Information Systems Requirements and Architectures with Business

agosto de 2017

An OMG Model-based Approach for Aligning Information Systems Requirements and Architectures with Business
Carlos Eduardo Rodrigues Teixeira Salgado
UMinho | 2017

Universidade do Minho
Escola de Engenharia

Carlos Eduardo Rodrigues Teixeira Salgado

An OMG Model-based Approach for Aligning Information Systems Requirements and Architectures with Business

Tese de Doutoramento
Programa Doutoral em Tecnologias e Sistemas de Informação

Trabalho efectuado sob a orientação de
Professor Doutor Ricardo J. S. M. Machado
Universidade do Minho, Guimarães, Portugal

Professora Doutora Rita S. P. Maciel
Universidade da Bahia, Salvador, Brasil

agosto de 2017

STATEMENT OF INTEGRITY

I hereby declare having conducted my thesis with integrity. I confirm that I have not used plagiarism or any form of falsification of results in the process of the thesis elaboration.

I further declare that I have fully acknowledged the Code of Ethical Conduct of the University of Minho.

University of Minho, ______________________________

Full name: **Carlos Eduardo Rodrigues Teixeira Salgado**

Signature:

—— This page intentionally left blank ——

Acknowledgments

To the colleagues in the ALGORITMI Research Center, especially those from the doctoral program and from the joint projects teams where we collaborated, kind gratitude for all the companionship, support and encouragement in this endeavor.

To the fellow researchers, speakers, reviewers and organizers of the conferences and related events which we took part on, sincere appreciation for all the sharing, experiences, comments and suggestions which contributed to the progress and growth of this work.

To the supervisors, priceless esteem for their timely monitoring, guidance, counseling and solidarity, in both high and low stages of this path. Even though all involving constraints of space and time, we managed to contact, discuss and exchange ideas when needed.

Finally, to family and friends, all thankfulness for the invaluable support, comfort and detachment provided. These conditions were indispensable for the necessary conciliation between personal, professional and research activities.

—— This page intentionally left blank ——

ABSTRACT

The challenges involved in developing information systems (which are able to adapt to rapidly changing business and technological conditions) are directly related to the importance of their alignment with the business counterpart. These challenges comprise issues that cross management and information systems domains, relating and aligning them in order to attain superior performance for the organization, while identifying its strategy and tailoring its business processes. As this relation is increasingly intertwined its concepts are conducted to pragmatic methods, incorporating both management and information systems components, for how, when and where this alignment really matters.

The related topics of the alignment between business and information systems comprise diverse paths of research, though with little common ground established inside the community, where problems arouse due to the fast moving business and technological environments. According to these circumstances, the process of developing information systems to support the alignment benefits from incorporating the use of structured and model-based approaches. So, as the development of evermore complex information systems presents a challenge for the currently available methods, the use of models to support the alignment with business stands as an increasingly important issue.

Following those challenges, we set out to question how to develop solutions aligning information systems with business in a model-based approach. Accordingly, we support our research on the need to understand what are the perspectives involved in aligning information systems with business, and, moreover, to comprehend in what sense model adoption drives information systems development. So, the proposed goals for this thesis are: (1) set the basis for the elicitation of business requirements in order to support a well-grounded development of information systems; (2) provide for the generation of business models based on the business requirements, while assuring their alignment and traceability; and (3) arrange for the derivation of information system architectures from the business requirements, while attaining alignment and traceability for their mutual transformation and adaptation.

Several issues surrounding these goals have already been described and approached in diverse ways by other researchers, where existing approaches and associated methods achieved good results. Nevertheless, these approaches are not without their shortfalls, sometimes failing to present a complete solution, others being unable to adapt to new challenges, or even incapable of reacting to recent trends. In order to tackle these issues we propose to build upon those approaches by adapting, evolving and innovating on solutions in each of the three proposed goals, respectively intertwining with perspectives from related standards and reference models.

Answering the first goal, in what regards the main contributions of this thesis, we propose to broaden the elicitation of requirements by relating functional and nonfunctional requirements from business processes. So, we present a unified metamodel representation for those requirements, accompanied by a customizable method for their joint elicitation, based-on business-driven use-cases, goals and rules. This approach adopts the Rational Unified Process (RUP) development methodology and the Business Motivation Model (BMM) standard model language representation for business requirements. Moreover, the metamodel representation and method operationalization are accompanied by a prototype support tool that completes this first contribution.

For the second goal, a more business-oriented one correlated to the higher-level requirements, we propose to generate business models directly from the inferred functional and nonfunctional requirements. So, we present a three-dimensional approach built on the relation of the referred requirements with the Balanced Scorecard (BSC) reference model, where an additional mapping to the Business Model Canvas (BMC) is also made available. This proposal provides an associated metamodel representation for the relation between the elements involved and a customizable method for their operationalization, all accompanied by a prototype support tool.

On the third goal, focused on system architectures and connected to the lower-level requirements, we propose to derive service-oriented participants from the functional requirements, while aligning the nonfunctional requirements with the quality characteristics of the solution to-be. First, we present an evolution of an existing method for the derivation of a logical architecture, in order to adapt it to a service-oriented approach (SOA). Then, following on the existing relation between the nonfunctional and functional side of the low-level requirements, our approach is able to associate these last with its related services on the derived architecture, in another three-dimensional approach. Additionally, a mapping of the nonfunctional requirements with the system quality characteristics (CISQ) is made available. Once more, an associated metamodel, a customizable method and a prototype support tool are also provided.

The development of these three approaches is supported through the execution of tasks which originate artifacts and lead to publications associated to their respective research and development efforts, all according to the Design Science Research (DSR) methodology. These are applied in ongoing projects involving experimental scenarios in industrial settings and associated to established research reference patterns, balancing the interests of both researchers and practitioners while focused both on technology and management audiences. The results obtained from their evaluation reflect the quality and depth of our findings, helping to validate the scientific contribution of this work.

Keywords: Information Systems, Requirements, Business Models, Architectures, Alignment

Resumo

Os desafios implicados no desenvolvimento de sistemas de informação (que sejam capazes de se adaptar a condições tecnológicas e de negócios em rápida mutação) estão diretamente relacionados à importância do seu alinhamento com a contraparte do negócio. Esses desafios envolvem questões que cruzam os domínios da gestão e dos sistemas de informação, relacionando-os e alinhando-os com o intuito de alcançar um desempenho superior para a organização, ao mesmo tempo que identificam a sua estratégia e adequam os seus processos de negócio. Como esta relação está cada vez mais interligada, os seus conceitos são canalizados para métodos pragmáticos, incorporando ambos os componentes de sistemas de informação e de gestão, para saber como, quando e onde este alinhamento realmente interessa.

Os tópicos relacionados com o alinhamento entre negócio e sistemas de informação abrangem diversos caminhos de pesquisa, embora com poucos alicerces em comum estabelecidos dentro da comunidade, onde os problemas surgem devido às rápidas mudanças nos negócios e nos ambientes tecnológicos. De acordo com estas circunstâncias, o processo de desenvolvimento de sistemas de informação para apoiar o alinhamento beneficia de incorporar o uso de abordagens estruturadas e baseadas em modelos. Assim, dado que o desenvolvimento de sistemas de informação cada vez mais complexos apresenta um desafio para os métodos atualmente disponíveis, o uso de modelos para apoiar o alinhamento com o negócio destaca-se como uma questão cada vez mais importante.

Em linha com esses desafios, estabelecemos a questão de como desenvolver soluções para alinhar sistemas de informações com o negócio numa abordagem baseada em modelos. Neste sentido, apoiamos a nossa pesquisa na necessidade de compreender quais são as perspetivas envolvidas no alinhamento dos sistemas de informação com o negócio, e, além disso, de compreender em que sentido a adoção de modelos capacita o desenvolvimento desses sistemas. Assim, os objetivos propostos para esta tese são: (1) definir as bases para o levantamento de requisitos de negócio a fim de suportar um desenvolvimento bem fundamentado de sistemas de informação; (2) disponibilizar a geração de modelos de negócio baseados nos requisitos de negócio, garantindo o alinhamento e a rastreabilidade entre ambos; e (3) estruturar a derivação de arquiteturas de sistema de informação a partir dos requisitos de negócio, preservando o alinhamento e rastreabilidade para a sua mútua transformação e adaptação.

Várias questões envolvendo estes objetivos foram já descritas e tratadas de diversas maneiras por outros investigadores, tendo as abordagens existentes e os métodos associados alcançado bons resultados. No entanto, essas abordagens têm as suas lacunas, umas vezes falham em apresentar uma solução completa, noutras são ineficientes ao se adaptarem a novos desafios, ou mesmo incapazes de reagir às novas tendências. Para lidar com estas questões, propomo-nos apoiar nessas abordagens, adaptando, evoluindo e inovando em soluções para cada um dos três objetivos propostos, intersetando-as, respetivamente, com perspetivas de modelos de referência e padrões relacionados.

Relativamente ao primeiro objetivo, no que concerne aos principais contributos desta tese, propomos alargar o levantamento de requisitos, relacionando os requisitos funcionais e não-funcionais dos processos de negócios. Assim, apresentamos um meta-modelo para a representação unificada desses requisitos, acompanhado por um método personalizável para o seu levantamento conjunto, baseada em casos-de-uso, metas e regras orientadas a negócio. Esta abordagem adota a metodologia de desenvolvimento do *Rational Unified Process* (RUP) e a representação padrão do modelo de linguagem do *Business Motivation Model* (BMM), para os requisitos de negócio. Além disso, a representação meta-modelo e a operacionalização do método são acompanhados por um protótipo de uma ferramenta de suporte que completa esta primeira contribuição.

Quanto ao segundo objetivo, mais orientado ao negócio e correlacionado com os requisitos de nível superior, propomos gerar modelos de negócio a partir dos requisitos funcionais e não-funcionais inferidos. Assim, apresentamos uma abordagem tridimensional, construída sobre a relação dos referidos requisitos com o modelo de referência do *Balanced Scorecard* (BSC), em que um mapeamento adicional para o *Business Model Canvas* (BMC) é também disponibilizado. Esta proposta inclui um meta-modelo para representação da relação entre os elementos envolvidos e um método personalizável para a sua operacionalização, tudo acompanhado por um protótipo de uma ferramenta de suporte.

No terceiro objetivo, focado em arquiteturas de sistema e ligado aos requisitos de nível inferior, propomos derivar participantes orientados-a-serviços desde os requisitos funcionais, alinhando os requisitos não-funcionais com as características de qualidade da solução a obter. Primeiro, apresentamos uma evolução de um método existente para a derivação de uma arquitetura lógica, adaptando-o a uma abordagem-orientada-a-serviços (SOA). Assim, prosseguindo a relação existente entre o lado não-funcional e funcional dos requisitos de baixo nível, a nossa abordagem associa estes últimos com os serviços relacionados na arquitetura derivada, numa outra abordagem tridimensional. Além disso, um mapeamento dos requisitos não-funcionais com as características de qualidade do sistema (CISQ) é disponibilizado. Mais uma vez, um meta-modelo associado, um método personalizável e um protótipo da ferramenta de suporte são disponibilizados.

O desenvolvimento destas três abordagens é suportado pela execução de tarefas, as quais dão origem a artefatos e levam a publicações associadas aos seus esforços de pesquisa e desenvolvimento respetivamente, tudo de acordo com a metodologia DSR. Estas são aplicadas a projetos em andamento, os quais envolvem cenários experimentais em ambientes industriais e associados a padrões de investigação de referência, equilibrando os interesses de investigadores e profissionais assim como dos diferentes públicos de tecnologia e gestão. Os resultados obtidos na sua avaliação refletem a qualidade e a profundidade dos nossos resultados, ajudando a validar a contribuição científica deste trabalho.

Palavras-chave: Sistemas de Informação, Requisitos, Modelos de Negócio, Arquiteturas, Alinhamento

Condensed Table of Contents

—— This page intentionally left blank ——

TABLE OF CONTENTS

—— This page intentionally left blank ——

LIST OF FIGURES

—— This page intentionally left blank ——

LIST OF TABLES

—— This page intentionally left blank ——

ACRONYMS / ABBREVIATIONS

IS - Information Systems

IT - Information Technologies

IST - Information Systems and Technologies

DSR - Design Science Research

RE - Requirements Engineering/Requirements Elicitation (where applicable)

BM - Business Models

EA - Enterprise Architectures

RUP - Rational Unified Process

4SRS - Four-Step-Rule-Set Method

MDA - Model-Driven Architecture

MDE - Model-Driven Engineering/Method-Driven Engineering (where applicable)

OMG - Object Management Group

BMM - Business Motivation Metamodel

SoaML - Service-oriented architecture Modeling Language

SPEM - Software Process Engineering Metamodel

CISQ - Consortium for IT Software Quality

SQC - Software Quality Characteristics

BSC - Balanced Scorecard

BMC - Business Model Canvas

SOA - Service-Oriented Architecture

PGR - Processes, Goals and Rules

—— This page intentionally left blank ——

CHAPTER 1

INTRODUCTION

Summary: This first chapter initiates with the context and motivation that led to exploring the research theme of this thesis. Afterwards, the research design is structured in three parts, starting by the formulated research question and associated goals, then the description of the followed research method (based on design science research) and lastly the activities planned for the time period prescribed. This chapter closes with an outline of the structure of this document and a synthesis of its contents.

CHAPTER TABLE OF CONTENTS

PUBLICATIONS:

- An OMG-based meta-framework for alignment of IS/IT Architecture with Business Models. (**QUATIC’14 [SEDES], IEEE**)

—— This page intentionally left blank ——

CHAPTER 1

INTRODUCTION

> *"Hope is like a path in the countryside. Originally, there is nothing — but as people walk this way again and again, a path appears."*
>
> – Lu Xun, **Chinese Essays** (1921)

1.1 CONTEXT AND MOTIVATION

Developing and maintaining an information system (IS) is one of the most important challenges an organization faces nowadays, claiming a large proportion of its resources. The lifespan and impact of information systems require appropriate methods for their creation, evolution and constant reengineering, in order to adapt to rapidly changing conditions, both business and technological ones [Banker, Davis, & Slaughter, 1998]. Poor design choices lead to misalignments in business/IS solutions, which are costly to support and difficult to change. Also, as complexity rises, the quality factors associated to these tasks become even more critical.

Alongside, research streams crossing management and information systems and technologies (IST) have been growing and maturing over the last years. This fosters the alignment between business and IS in what regards the managerial problems that relate to IS, the methodological paradigms to study them and their impact in managerial practice [Banker & Kauffman, 2004]. Among others, IS supports business in organizations through systems analysis and design, management of software and IST investments, configuration of business processes and formation of business strategies. It also helps different stakeholders to understand, adapt to and effectively manage technologies.

Attaining the necessary performance requires alignment between the organizations strategies and the business processes, where alignment is especially relevant on software development processes. Also, the alignment choices on products characteristics, customers and business strategies must be considered, all being matched and customized to complement different types of product requirements and strategies [Slaughter, Levine, Ramesh, Pries-Heje, & Baskerville, 2006]. One of the challenges is to identify the dimensions that influence these within the business unit strategy, where alignment is done, for then tailoring the operational processes accordingly.

Historically, the alignment between business and information technology (IT) had a stronger approach on the managerial and societal side, with managers and researchers trying to find the secret potion of alignment through measuring organization performance, value creation, critical success factors and others. These views originated many stranded paths of research, with little common ground established inside the community, which led to scarce and feeble advances [Henderson & Venkatraman, 1993]. Also, although alignment is currently the dominant term in literature, other synonymous are used, like "fit", a term with much history and associated research work, namely on the mathematics and strategy domains.

On a more holistic and normative sense, the Strategic Alignment Model (SAM) presented itself as a first inspiration where to guide our work on, with its original model later defined and reviewed in a more practical way [Luftman, Papp, & Brier, 1999]. Also, it was further enhanced and refined through a unified framework for interrelating the different components of information management, dealing with business, information, communication and technology at the strategic, structural and operations levels [Maes, Rijsenbrij, Truijens, & Goedvolk, 2000].

It was the first to transform the concept of alignment into a practical method, incorporating both management and design components. By giving SAM an empirical support and a more robust framework (Fig. 1.1), it provided managers and researchers with additional practical ways to attain and monitor alignment, while changing strategy accordingly [Avison, Jones, Powell, & Wilson, 2004]. Albeit this was an isolated study, it contributed in filling gaps to understand how alignment is actually achieved in organizations in a dynamic environment. Accordingly, similar research efforts on longer periods and with a wider range of scenarios are required.

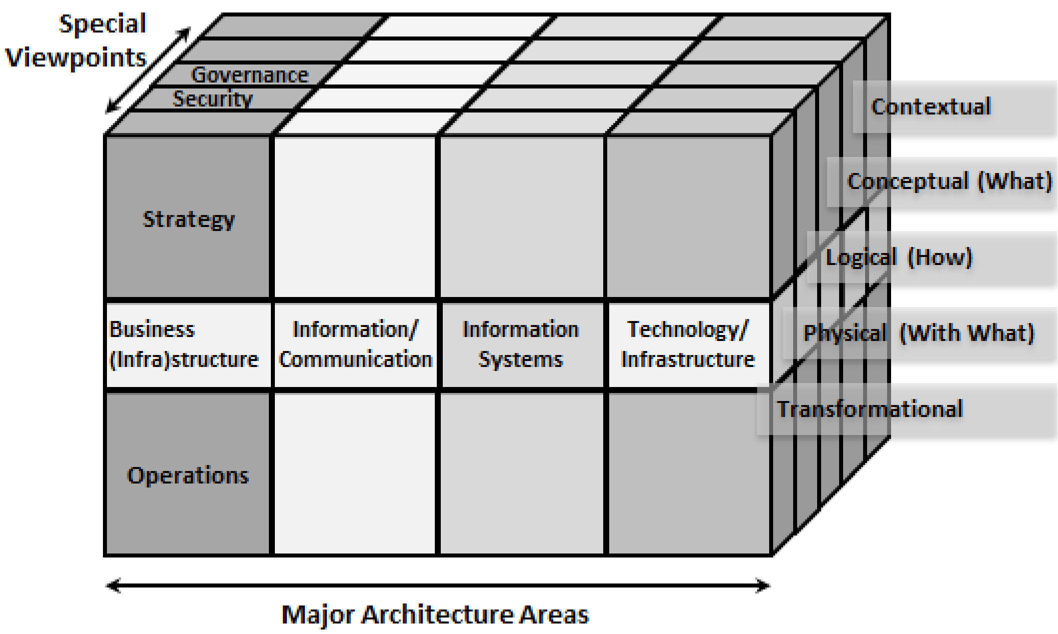


Fig. 1.1: Alignment through a unified framework (adapted from [Maes *et al.*, 2000])

With the increasingly higher interest in this topic, a thorough state of the art on IT alignment laid the foundations for much answers [Chan & Reich, 2007]. Meanwhile, it left many open roads for research (like questions to go beyond "alignment is good" statements) and explored details concerning how, when and where IT alignment really matters. Alignment is presented as a non-static and non-single-dimension process, which not being easy to attain, often leads to complex and chaotic situations that should be regarded as an ongoing process with no end state.

Other complementary topics of interest range from levels of alignment, strategy on the upper level of the organization and operations goals on the frontline, to ways of assessing and measuring alignment, establishment of typologies and taxonomies, congruence of business and IT strategies, and quantitative and qualitative descriptors. Notwithstanding, ever new problems are raised due to the fast moving business and IST environment.

Our interest in business/IS alignment comes from the fact that organization management and IST are increasingly intertwined nowadays. Evermore, organizations needs to use IS to achieve better performance, knowledge and communication overall, and to manage its statistics, measures and levels of service. Consequently, this makes the alignment between IS and business one of the top concerns of people in organizations, and we feel it too in our daily work. Organizations, people, processes, IST and the entire surrounding environment need alignment for a better integration and agility in working together!

As alignment is a complex reality with no agreed-upon definition inside the business and IS realms, a question that we do not intend to pursue, we see it as much more than just handling customer data and share the vision that one way of achieving alignment is to develop IS that provide services for a business to meet its needs (Ullah & Lai, 2013). Also, we intend to move away from the too theoretical processes and use of unclear business goals, as existing in the historically business-driven approaches. Instead we focus on methodologies directed to IS development, which should be also prepared to monitor and evaluate the proposed solutions in order to act accordingly along its evolution. These methodologies must effectively link business and IS, counting on important pillars as business models, business strategy and operations.

This requires agile handling of requirements engineering (RE) and design of enterprise architectures (EA), in shorter timeframes, while taking into account the organization environment. Consequently, its associated development needs an appropriate RE process, supported on early goal modeling, flexible to manage the occurring rapid changes and recurring to the use of models as the involved techniques are usually complex (Ullah & Lai, 2011). Also, EA is essential as it can support alignment, especially as a tool, and aid in understanding IS properties when managing and analyzing alignment, in a process where situation-specific, adaptable solutions are needed and not a one size fits all solution (Saat, Franke, Lagerström, & Ekstedt, 2010). Moreover, this strengthens the importance of using metamodels to provide a common language, and the need for a prescriptive perspective directed to the construction and delivery of artifacts.

Furthermore, it is important to set our terms clearly and fight ambiguity throughout our research work. One example is the case of information systems (IS) and information technologies (IT), which are related but different terms, although some people do not make a distinction. Even that no agreed-upon definition for the field of IS exists, some important references have been set, either regarding IS in a dual perspective of the observed system, and the organization and activities of the IS function [Davis, 2000], or in contemplating the view of IS as a work system [Alter, 2008].

In our understanding, IS refers to systems designed to create, store, manipulate or disseminate information, even in the extreme case of no technology being involved, while IT is the branch of engineering that deals with the use of software, computers and communications, the technologies adopted to implement the IS. Throughout academic and technical research, not even mentioning the day-to-day practice in business organizations, the terms IT and IS are used with poor rigor, so we propose to use only IS, as in our view the IT is simply a part of IS. Nevertheless, we are aware that this decision faces some resistance, both in academic and practice communities, as IT is already deeply engraved in the mind of researchers.

Additionally, as the usage of standard methods and models are an essential part of our work and vision, based on our background training on mathematics and computer sciences, the theme and aim of this thesis is purposely influenced by the Object Management Group (OMG) specifications, and oriented to the development of model-based and methodological-oriented solutions. Besides our particular interests on the use of models, these play an important role in the context of IS development, with the added advantages of using models for business/IS alignment. Moreover, model-based solutions are considered a good strategy to achieve alignment, although performing this alignment is very challenging because of all the issues involved.

The challenges surrounding the business/IS aligning, the need to ground our work on more standardized methods and the focus on development of model-based solutions, all received good acceptance inside the Software Engineering and Management Group (SEMAG)[1] of the ALGORITMI Research Center at the Universidade do Minho (UM). It was then natural that the work that led to this thesis would be framed inside our participation in the Doctoral Program in Information Systems and Technologies (PDTSI), in direct connection with the Information Systems Department (ISD) of the UM, at Guimarães, Portugal. Also, in search for collaboration in the referred issues, the Centro de Computação Gráfica (CCG)[2], at Guimarães, Portugal, and the Computer Sciences Department (DCC)[3] of the Universidade Federal da Bahia (UFBA), at Salvador, Brazil, arose as interested partners on joint projects and research activities. The work associated with this thesis is then undertaken within SEMAG and the ALGORITMI Research Center, with close collaboration from CCG and the cooperation of DCC-UFBA.

Finally in this endeavor, as the scientists of the first days of computer science, we will be doing Design Science Research (DSR) [Hevner, March, Park, & Ram, 2004]. As one of its guidelines is on building on existing knowledge, our decision also concentrates in working on previously existing solutions and towards some consensus, in order to mitigate the dispersion and fuzziness of the alignment between business and IS topics, and avoid the proposal of standalone solutions detached from the research and practice communities.

1.2 RESEARCH DESIGN

Our research design comprises the definition of the expressed research question and its associated research goals, added to the foundations, description and adaptations of the followed research method. Moreover, the exposed plan for the list of tasks and resources involved in all of the research activity for this thesis is also included in this section.

We first address the purpose of research, i.e., what is the main question addressed, and what are the issues raised by its derivative goals and the expected contributions from following on each one of them, thus presenting the objectives of this research and its intended results. Next, we describe the approach which is used to explore the question and goals set, and explain how the research

[1] http://algoritmi.uminho.pt/research-teams/semag/
[2] http://www.ccg.pt/en/
[3] http://wiki.dcc.ufba.br/DCC/

method is adapted and used to validate our findings. Among the activities planned for this thesis we encompass the envisioned partnerships and projects, the preparation and revision of documentation and articles, and the development of tools support, alongside the conferences to submit articles for publication and presentation.

Research Question and Goals

A schematic of our research problem is depicted in Fig. 1.2, where we start by defining the research question, which guided a state of the art for this work, in the form of a literature review. Associated to the main research question (RQ), we set three related research goals (RG) which are analyzed throughout this document, respectively, in each dedicated contribution chapter. Individual research goals are then followed through the execution of tasks, which originate artifacts and lead to publications associated to their respective research and development efforts.

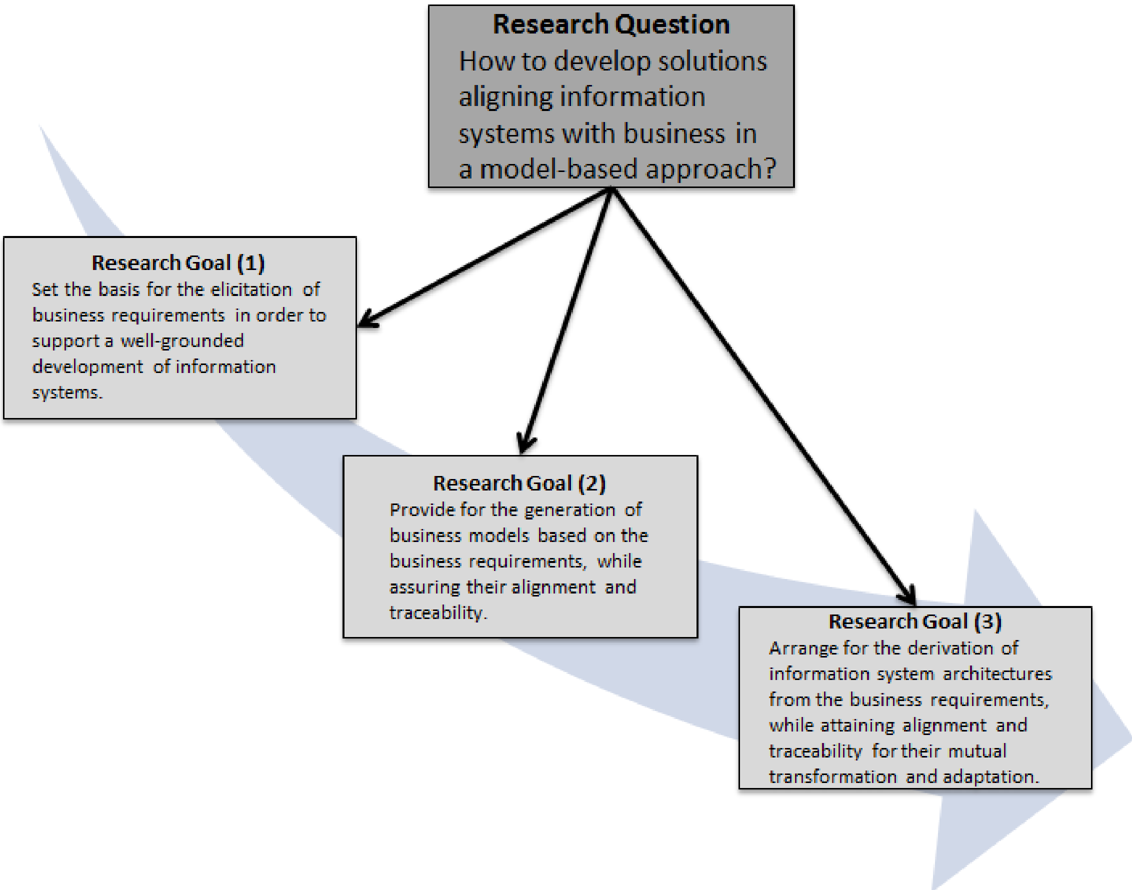


Fig. 1.2: Research question and associated goals

RQ: How to develop solutions aligning information systems with business in a model-based approach?

Developing successful and evolvable information systems is a complex task, requiring appropriate methods that adapt to rapidly changing conditions and attain superior performance. Also, on the search for a strategic and operational alignment between business and information systems, the correct use of proper terms and models is of the essence. These issues are major hurdles in the development of any project, where currently available methods are not fully capable to embrace them, as further detailed in the state of the art chapters of this document. So, our research question stands on the quest to deliver methods that allow developing model-based solutions aligning information systems with business.

As business/IS alignment is both the domain and a central point in the research question of this thesis, one could consider this later composition too generic or possibly vague, nevertheless, by restricting the focus of the research question it would undermine its transversal nature. This is stated by the broad coverage of topics in the process of developing information systems solutions, and added to the use of tailorable and cohesive model-based approaches. Also, in what regards the research goals, it must be taken into account that their somewhat wide scope is then narrowed in each of the contributions made, according to their respective contexts. Another hurdle in presenting a more focused and clearer problem statement, which could synthetize holes in business/IS alignment and complement other existing approaches, is that, as to the best of our knowledge, there is none that integrates the requirements, business and architecture interrelated concerns.

Alongside this question, the need to understand what perspectives are involved in aligning information systems with business and its associated topics, and, moreover, to comprehend in what sense model adoption drives information systems development, are also under consideration. The first suggests that alignment between business and information systems comprises several domains, crossing perspectives on different topics and issues within those domains. More specifically, we turn our attention to the initially identified topics of requirements engineering, business models and enterprise architectures. The importance of model adoption is a second-side issue related to our research question, involving the questions of formality, standardization and customization, especially in what regards the information system basis for the development of methods and tools support. Also associated to this last, the questions related to the use of standards and associated specifications need to be contemplated.

RG.1: Set the basis for the elicitation of business requirements in order to support a well-grounded development of information systems.

Requirements engineering is a central and essential piece on the alignment topic, traversing the business model and enterprise architectures domains. Outlining the type of requirements needed to address the current quality challenges for development of the intended solutions and delineating a method for their successful elicitation is the first goal in our research, which also sets the basis for the following two aligning goals.

RG.2: Provide for the generation of business models based on the business requirements, while assuring their alignment and maintaining traceability.

Although some steps have been taken in that direction, the research community still has no consensual definition and representation for a business model concept, so which one should be used? Also, how can a business model be generated by following the initial elicitation of requirements? More even, how to trace back and impact those requirements from any changes happening in the business model? These are all challenges related to this second goal.

RG.3: Arrange for the derivation of information system architectures from business requirements, while attaining alignment and traceability for their mutual transformation and adaptation.

In a more 'lower-level' alignment direction, the classic subject of the transformation between requirements and architectures stands as a long thought problem. Among the midst of transformations available, still much work is needed in order to advance the research for this topic. The main goal lies in the kind of relation between requirements and architectural issues, maintaining traceability between elements from both sides and on the process to go back and forth the two representations, reflecting changes and adapting mutually to one another.

Research Method

The decision for concentrating our work on previously existing solutions and towards some consensus follows from one of the previously referred DSR guidelines for building on existing knowledge. This choice also allows mitigating the dispersion and fuzziness associated to the researched topics. Moreover, as our early state of the art identified the problems addressed on this current study, most of which were already described and approached in diverse ways by other researchers, the goals associated to the research question are appropriate to be pursued through DSR. For as it "seeks to extend the boundaries of human and organizational capabilities by creating new and innovative artifacts" [Hevner *et al.*, 2004], and these "artifacts are broadly defined as constructs (vocabulary and symbols), models (abstractions and representations), methods (algorithms and practices), and instantiations (implemented and prototype systems)".

This due, our proposed study is structured according to the DSR methodology, addressing important unsolved problems in unique or innovative ways, or solving problems in more effective or efficient ways. There, the key differentiator between routine design and design research is the clear identification of a contribution to the archival knowledge base of foundations and methodologies. Moreover, the existence of three design science research cycles (Relevance, Design and Rigor) in any design research project [Hevner & Chatterjee, 2010], puts in evidence the relation of Design Science with the Environment and the Knowledge Base (Fig. 1.3).

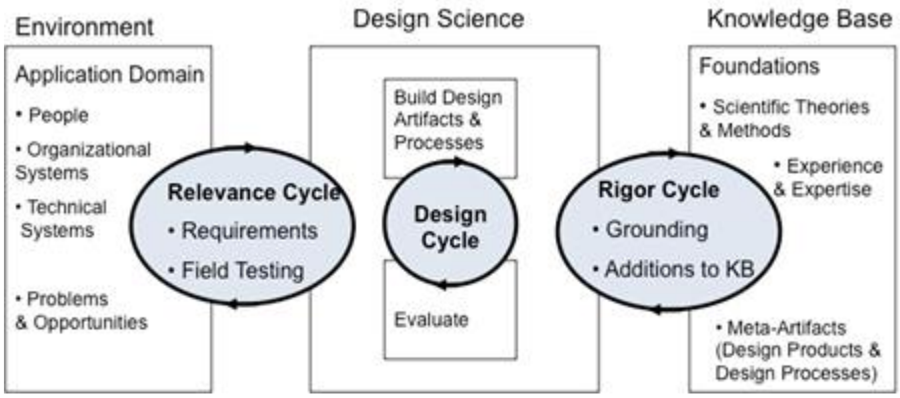


Fig. 1.3: Three research cycles [Hevner & Chatterjee, 2010]

First, the Relevance Cycle relates the contextual environment of the research project with the DSR activities. From the application domain Environment, we obtain an identification of opportunities and problems, the requirements for the research, and also the acceptance criteria to evaluate research results. Then, the Rigor Cycle relates the DSR activities with the Knowledge Base of scientific foundations that grounds the research project. This cycle execution avoids that the technological artifacts become the only focus without considering an adequate theory base, potentially resulting in well-designed artifacts useless in real organizational settings. Finally, the center of DSR, the Design Cycle iterates on building and evaluating the design artifacts, in so enriching the Design Science repository. The activities of building an artifact, evaluating it and refining its design, are realized more frequently than activities included on the other cycles.

In line with DSR, we will follow the view for excellence proposed in the form of seven guidelines useful for understanding, executing and evaluating design science [Hevner *et al.*, 2004]:

1. Must produce a viable artifact.
2. Produces technology-based solutions to relevant business problems.
3. Evaluation that demonstrates of utility, quality, and efficacy.
4. Research contribution of the design artifact, foundations, or methodologies.
5. Rigor in construction and evaluation method.
6. A problem-situated means-ends search for an effective artifact.
7. Communication to both technical and managerial audiences.

Related to the DSR thematic, another relevant point in these last years was the publishing of DSR articles and the distinction between journals with technology-focused and management-focused audiences. As it is true that good DSR should produce results of interest for both audiences, in our point of view we will try to give technological audiences sufficient detail to enable the described artifact to be implemented and used, and to management people sufficient detail to determine if organizational resources should be committed to acquiring and using the proposed artifacts, all within each specific appropriate context.

Concurrently, IS research is also criticized for having little influence and low relevance on practice. One way to obviate to this is to conduct research using appropriate research methods that balance the interests of both researchers and practitioners, being these two approaches, action research and design science research, a good example. Their combined analysis revealed interesting parallels and similarities between the two, and proposed ways to facilitate their integration [Purao, Rossi, & Sein, 2010].

The process models of both approaches were found to be similar to a degree that a common process model can be formed and outlined in an integrated approach. Also, as both approaches have common starting points and goals, the choice between design science research and action research can be done at the stage of the research where the intervention is planned. This finding allows us to start our work with design science research and conveniently commute to action research when associated projects call for a more active participation, helping in achieving and validating results.

Research Activities

Our research is distributed into three main segments, besides the initial literature review and base thesis writing itself, each of them relating to an answer to a subsidiary research goal which originates the respective artifacts to be produced. Planned to last a total period of around four and a half years (52 months, 224 weeks or 1550 days), from September 2011 to December 2015 (Fig. 1.4), the initial chronogram is prepared to face some fluctuations that occur during this period, nothing uncommon due to the iterative and dynamic nature of the research process. This follows accordingly to the doctoral program organization, where a curricular year precedes the thesis phase in order to produce an early state of the art and prepare the research plan.

This thesis explores issues on the alignment between business and information systems, where the early studies indicated that this is a clearly fragmented topic, with portions from strategy, through business models, goal modeling, enterprise architecture and requirements engineering. After the categorization and analysis of the results from the early state of the art, three main aggregation topics were uncovered, namely: requirements engineering, business models (including the topics of business strategy and goal modeling) and enterprise architectures, these last two with transversal references to requirements engineering. In all of them there is still much fuzziness, due to their relatively recent research history and lack of consensus within each diverse proposals and trends, nevertheless, as their past history and current efforts are scrutinized, we identify future tendencies and research topics for each of them ahead in the state of the art chapters of this document.

Our research activities are concerned with tackling these issues, especially on the comprehensive elicitation of the different kinds of requirements for the development of information systems. From these, we set to determine the constituents of an associated business model and their relation with the previous requirements. Again based on these last, the transformation between requirements and architectural elements, as well as the widespread benefits of enterprise architecture solutions, are major concerns of our research.

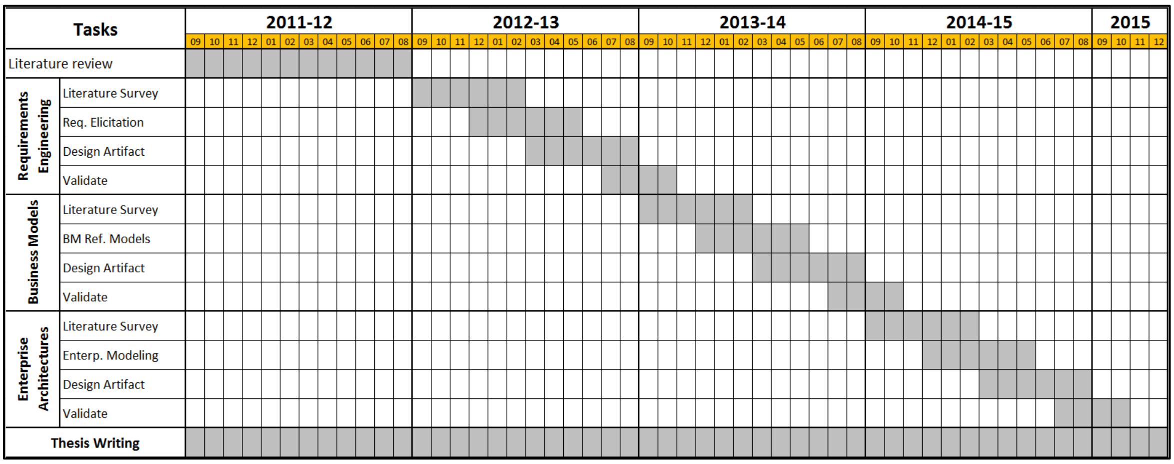


Fig. 1.4: Research execution initial chronogram

The associated tasks in all of the three segments contribute for updating and broadening the literature review on each of the topics. On one hand they support the development of the proposed work, while in the other continue to develop the state of the art, so feeding updates from the thinner research associated to the branches of each segment. There are some planned conference articles associated to each of the three segments, in so helping to validate and share this work with the research community. The acceptance and publication of papers on conferences also aims to validate the quality of our research and respective articles, receive important feedback from the research community, as well as to give visibility to our scientific contribution and gather public recognition for the developed work.

Periodically publishing our findings in recognized scientific publications, through the participation in conferences, also allows us to interact with the scientific community and exchange knowledge and ideas, which constitutes a very motivating experience. Also in our plans is the publishing of longer articles in renowned journals in the area, nevertheless, due to the time restrictions of this thesis and the long publishing peer review process associated to them that is currently out of scope for this document. But, as the end of this work approaches, some contributes of this thesis will be prepared and submitted for these type of publications.

Additionally, we count on other methods in order to further evaluate and validate our findings, namely through the participation in projects, involving experimental scenarios in industrial settings and associated to established research reference patterns. This is implemented in the form of ongoing projects inside the ALGORITMI Research Center, which brings a more practical orientation to our research and improves the quality and depth of our findings. These are important opportunities for us to apply and test our studies, with a chance to embrace some tasks in action research for a more practical and hands-on approach.

The established collaborations and the way in which the associated tasks are prosecuted aim to contribute both with concrete results to the projects and in advancing our own research status. The research work is performed mainly within the scope of the AC.pt and AAL4ALL projects, wherever our work can contribute in a positive way, but also with minor involvements on other closing projects, as the ISOFIN and SLVCement, and in new projects to be launched, as the CloudAnchor. Our participation consists in performing the role of either an advisor, recommending directions to follow and reviewing the range of available solutions, or as a contributor, providing for the development of specific tasks and creation of needed artifacts. The AC.pt and AAL4ALL projects are used as demonstration cases within the business model and architectural services developed solutions, respectively in chapters 5 and 6.

Most of these projects are under the responsibility of our SEMAG research group, where the V-Model approach [Ferreira, Santos, Machado, & Gašević, 2013], alongside the four-step-rule-set (4SRS) method [Machado, Fernandes, Monteiro, & Rodrigues, 2006], are widely used for their development process, especially in ill-defined contexts. Albeit process-oriented approaches as the V-Model allow eliciting requirements and obtaining a process-view of an information system, achieving good results, much information is lost by not uncovering the business vision and strategy information artifacts. As this handicaps the obtained solution and its future evolution, our work intends to work upon and evolve the methods inside the V-Model, namely by further aligning the existing metamodels while integrating others, to continually improve its current features and bringing new ones to strengthen the existing solutions.

1.3 STRUCTURE OF THIS DOCUMENT

This document is structured in seven chapters, each one of them preceded by a cover page with a short summary, an index of its contents and also the list of associated achieved publications that contributed to each specific chapter. This organization intends to provide a quick grasp on the contents of each chapter, and assist on its perception and impact. Additionally, the entire set of references used throughout this document is aggregated in a final chapter at the end of this document.

With the exception of the first and last chapters, the main sections inside the chapters are systematically organized with an initial introduction and an end conclusion, and, in between these, the in-depth sections for the chapter's theme. Especially on the contributions chapters, these sections are organized in growing complexity, starting with an initial approach on its related topics and a first proposal to tackle them, followed by a further developed and innovative solution from our part, and end with the description of an associated tool support, essential to visualize and fully grasp the added contributions, as well as for evaluating their feasibility.

The seven chapters of this document and their main content are, respectively:

1. **Introduction** – this first chapter presents the context and motivation for this thesis, addresses the research question and associated goals, alongside the research method followed and the activities planned for the duration of the doctoral program time span, and also the structure of chapters of this document.
2. **Perspectives on Aligning Information Systems with Business** – following on the issues aroused by the research question, this second chapter takes a look into three identified perspectives on aligning information systems with business, by surveying the topics of requirements engineering, business models and enterprise architectures, while detailing some of their related issues.
3. **Adoption of Models in Information Systems Development** – relating to a second dimension of the research question, this third chapter surveys the modeling and methodologies issues associated to information systems development, presents the OMG specifications and corresponding metamodels that matter to this thesis, and browses the current available options for support tools development environments.
4. **Elicitation of Business/IS Requirements: The PGR Approach** – pursuing the first goal and as a first contribution, this fourth chapter sets the stage for the focus of our research work; first presents a comparison between our chosen 4SRS method with other related transformation methods between requirements and architectures, and then advances with a proposed approach for the elicitation of functional and nonfunctional requirements, directly from business processes, goals and rules (PGR), accompanied by the specification of a metamodel and a method, and the corresponding development of an associated prototype tool support.
5. **Generation of Business Models from PGR-based Requirements** – in a second contribution directly related to the previous one, this fifth chapter covers the topic of business models; starts with an initial method for the generation of a business model canvas through reengineering the elicitation of business processes, goals and rules, and then upgrades it by adding BSC as a third dimension to the functional/nonfunctional duality, in a cube-like solution; all this is equally accompanied by the specification of a metamodel and a method, and the corresponding development of an associated prototype tool support.
6. **Derivation of PGR-based Service-Oriented Architectures** – as a third and final contribution, aimed at its respective goal and closing the circle on the information system alignment with business, this sixth chapter details the evolution of the 4SRS method to handle the derivation of a service-oriented architecture, with the added concern on quality issues; again building on the dual functional/nonfunctional requirements, with the quality information associated to services as a third dimension, this second cube-like solution is once more accompanied by the specification of a metamodel and a method, and the corresponding development of an associated prototype tool support.
7. **Conclusion** – this final chapter concludes this document, by synthetizing a critical analysis of the performed work, and also a resume of the contributions in the different topics identified, accompanied by the listing of the publications achieved along the research chronogram; finally some of the limitations involved, along with some lessons learned, and some hints for future work are presented.

—— This page intentionally left blank ——

CHAPTER 2

PERSPECTIVES ON ALIGNING INFORMATION SYSTEMS WITH BUSINESS

Summary: As a base state of the art in the form of a literature review from where to build on, this chapter covers three perspectives directly related to the thematic of aligning information systems with business. Initially it browses through requirements engineering, as the fundamental, transversal topic that connects all perspectives. Following, it surveys from the business models, a still ambiguous topic but with already solid references available, until the enterprise architectures, focusing on the enterprise modeling, goals, strategy and services related issues.

CHAPTER TABLE OF CONTENTS

PUBLICATIONS:

- Modeling the Alignment Between Business and IS/IT: A Requirements Engineering Perspective (**SAC'13, ACM**)

—— This page intentionally left blank ——

CHAPTER 2

PERSPECTIVES ON ALIGNING INFORMATION SYSTEMS WITH BUSINESS

"So the problem is not so much to see what nobody has yet seen, as to think what nobody has yet thought concerning that which everybody sees. Also for this reason, it takes so very much more to be a philosopher than a physicist."

– Arthur Schopenhauer, **Parerga and Paralipomena: Short Philosophical Essays** (1851)

2.1 INTRODUCTION

The alignment between business and information systems has a relatively recent and somewhat short history in the field of IS research, nevertheless a not so uninteresting one. Subsequently, as this topic has been for already some time a top concern of researchers and practitioners in diverse areas, leading to much dispersion and fuzziness along the way, an early literature search on its concepts was necessary.

Although no specific literature review protocol was followed, we performed thorough searches on the Web of Knowledge[4] and Google Scholar[5] academic search engines. They covered different time spans, with search terms composed of keyword variations and synonyms on business, alignment, IS, IT and related words of requirements engineering, enterprise architecture and strategy alignment. English was the language of choice, accounting for the totality of selected references, which were designated and cataloged according to their relevance, number of cross-references, date of publishing, related work of authors and type of studies.

After our literature search, three main topics of study were identified to take part on the state of the art. These were the requirements engineering, with a special focus in the activity of requirements elicitation, the business model artifact, covering the popular balanced scorecard and business model canvas references, and the enterprise architecture, in particular regarding the solutions related to enterprise modeling and services. Besides the contents from the initial research effort,

[4] https://www.webofknowledge.com/

[5] http://scholar.google.pt/

this state of the art chapter also counts with contributions from the continued research work, namely from the background sections of paper publications, on each of these topics throughout the time span of this thesis.

Requirements engineering (RE), sometimes disguised as one of its tasks: elicitation, modeling, analysis, validation and verification, and management, is referred across all the topics as it has strong connections to them. So, first we perform an exploration into the topic of requirements engineering, with different, and sometime provocative, views on problems and solutions, and then dive into its details, browsing the more traditional functional side and analyzing the not so considered nonfunctional part.

On another topic, after a somewhat long history associated to business practice, the business model (BM) is nowadays considered a central point in the alignment of an organization, traversing through its strategy, goals, requirements, until its IS model and IT architecture. Then, besides describing this evolution and the fuzziness still surrounding this topic, the evermore relevant paper that the balanced scorecard (BSC) and business model canvas (BMC) play as central references on this topic is also analyzed.

Finally, a similar approach is performed on the topic of Enterprise Architecture (EA), covering research from the early days to recent work and future tendencies. Several tentative reference models and frameworks have been proposed, trying to answer the needs of this central topic in the business/IS alignment. Conversely, enterprise modeling has evolved alongside it, with related research issues around goals, architecture, business models and services.

2.2 REQUIREMENTS ENGINEERING

Requirements Engineering (RE) has many facets, but its main concern is the process of finding a solution for some problem. It can be approached from a problem-oriented view, which focuses on understanding the actual problem, or from a solution-oriented view, focusing on the design and selection of solution alternatives [Wieringa, 2004]. In this respect, three orientations have been described: problem-oriented, solution-oriented and problem chains [Engelsman, Quartel, Jonkers, & van Sinderen, 2011].

Problem-oriented views are based on investigation and documentation of the problem domain, building a requirements model that describes the experienced problematic phenomena and the relations within. A popular problem-oriented approach is goal-oriented RE (GORE), with well-known GORE techniques as i* [Yu & Mylopoulos, 1994] and KAOS [Dardenne, Van Lamsweerde, & Fickas, 1993]. Here, goals are considered as high-level objectives of some organization or system, capturing the reasons why it is needed and guiding decisions at various levels within the enterprise. Its use facilitates reasoning about the purpose of a proposed solution, as goal models can be analyzed to demonstrate which ones realize the others, and which ones conflict or negatively contribute to others, while demonstrating the contribution of the proposed solution to the actual need.

Solution-oriented represents the more traditional software engineering view, where a requirements model typically describes the context of the system to-be, the desired system functions, their quality attributes and alternative configurations, or refinements of these functions and attributes. These alternatives are then analyzed and compared to decide which one is the best solution to the problem. Traditional approaches are structured analysis and object-oriented analysis, which focuses on the flow of data and control of the system to-be, and applies object-modeling techniques to analyze the functional requirements of the system to-be, respectively. An important object-oriented analysis technique is use-case elicitation and specification [Jacobson, Booch, & Rumbaugh, 1999], with use-cases capturing the solution behavior in terms of interaction scenarios between the system and its user.

The problem chains, where each chain links a problem to a solution such that this is considered again as a problem by the next chain, link RE to Enterprise Architecture (EA). The "why" represents the problem-oriented view and defines the business needs, goals, requirements and use-cases that should be addressed, and "what" represents the solution-oriented view in terms of EA elements such as services, processes and applications. These architecture elements define what the enterprise must do to address the business needs, and at the same time, motivate and justify why the EA is defined the way it is. This way of thinking enables traceability, as EA can be traced back to the goals and requirements that motivated their introduction, while RE can be traced forward to services, processes and applications that implement them. This allows analyzing and managing the impact of changes to an enterprise.

RE is also the process of discovering if an information system, many times in the form of a software system, meets the purpose for which it was intended, by identifying stakeholders and their needs, and documenting these in order to proceed to analysis, communication and subsequent implementation [Nuseibeh & Easterbrook, 2000]. The needs or goals of the stakeholders may vary and conflict, and depending on their perspectives, may not be explicit or may be difficult to articulate, which may constrain their realization. RE has a comprehensive array of activities, from elicitation to modeling and analyzing, and communicating, among others.

Regarding elicitation, there should be taken in account the need for the information gathered to be interpreted, analyzed, modeled and validated before the requirements engineer can feel confident that a complete set of requirements has been collected. The requirements to elicit depend on what problem needs to be solved, the identification of stakeholders, goals denoting the objectives a system must meet, tasks users currently perform, or use-cases and scenarios, all depending on the approach chosen. Several elicitation techniques are available, also depending on each situation, be it traditional, through questionnaires, surveys, interviews or analysis of existing documentation, with group elicitation or using brainstorming and focus groups, as well as Rapid/Joint Application Development (RAD/JAD) workshops, prototyping, model-driven, through goal-based methods and scenario-based methods [Nuseibeh & Easterbrook, 2000].

Modeling and analyzing requirements is nowadays a fundamental activity in RE, as well as the modeling methods and their associated analysis techniques. One of the fields involved is enterprise modeling, often used to capture the purpose of a system by describing the behavior of the organization, which is expressed in terms of organizational objectives or goals, and associated tasks and resources. Also, through the enterprise business rules, workflows and the services that it will provide, the high-level business goals can be refined repeatedly as part of the elicitation process, leading to thinner requirements.

Communicating involves the process of facilitating effective communication of requirements among different stakeholders. Here, requirements traceability represents another major factor that determines how easy it is to read, navigate, query and change requirements documentation, being a mean of achieving integrity and completeness of documentation, and with an important role to play in managing change. Accordingly, a common practice is to devise experiments to attempt to refute the current statement of requirements, which shifts the problem of validating requirements statements to a problem of convincing stakeholders that the chosen representation for requirements models is appropriate. Also, as systems evolve as the environment in which they operate and its stakeholder requirements change, the automated support for propagation of change becomes crucial.

Furthermore, RE as a multi-disciplinary activity deploys a variety of techniques and tools at different stages of development and for different kinds of application domains. Existing methods provide a systematic approach to combining different techniques, notations, heuristics and guidelines for the requirements engineer to deploy at different stages of the process, or at a particular problem or domain. Also, it is common agreement that for the effective management of an integrated RE process, automated tool support is essential.

Research in RE is constantly the target of updates and reviews, with future research directions in the field being identified and specific requirements tasks addressed, such as elicitation, modeling and analysis [Cheng & Atlee, 2007]. Nevertheless, there is a common consensus on considering requirements to reside primarily, but not exclusively, in the problem space whereas other system and, specially, software artifacts reside primarily in the solution space. However, one is always crucial in influencing the other, as when relating requirements and architecture there is no fundamental distinction between architectural decisions and architecturally significant requirements [de Boer & van Vliet, 2009].

In RE the prevailing view tends to be the requirements engineering as problem analysis, whether in the architecture view, it is the architecture as solution structure. Notwithstanding, the choice for a particular solution may introduce new problems and hence new requirements, that is, architectural design decisions may introduce new problems to be solved by subsequent decisions.

Also, as RE is constantly evolving, evermore there are implications of new principles, counting on new practices and realities as the integration of enterprise-level and nonfunctional requirements. These new ways approach requirements design with key principles as intertwining requirements and contexts, and evolving designs and ecologies. By managing through architectures and recognizing complexity, they move towards evolutionary platforms defined by complex consumption and production realities [Jarke, Loucopoulos, Lyytinen, Mylopoulos, & Robinson, 2011].

2.2.1 Requirements Elicitation

The activity of elicitation, including the selection of which technique, approach or tool to use, is dependent on a large number of factors including the type of system being developed, the stage of the project and the application domain, to name only a few [Zowghi & Coulin, 2005]. Because of the relative strengths and weaknesses of available methods, and the type of information they provide, a combination of several different techniques is necessary to achieve a successful outcome. Despite attempts to automate parts of the process, and develop frameworks and guidelines, requirements elicitation still remains more of an art than a science.

Models, like use-cases, scenarios, enterprise models, and some policy and goal models used during elicitation, tend to be informal and intuitive to facilitate early feedback from stakeholders [Cheng & Atlee, 2007]. Also, they tend to be more precise, complete and unambiguous, so to be used to communicate the requirements to downstream developers. This helps to raise the level of abstraction in requirements descriptions, by providing a vocabulary and structural rules that more closely match the entities, relationships, behavior and constraints of the problem being modeled. Usually, scenario-based models are the preferred focus of researchers, as they are easier for practitioners and nontechnical stakeholders to use.

Nonetheless, it is virtually impossible to define a unified model for the elicitation process, due to the constantly changing needs associated to requirements activities. Even if specific methodologies, broken down into multiple steps, describe general approaches and overall principles to assist analysts in understanding needs, only the experienced analyst understands intuitively which method or technique is effective, in each circumstance, and is able to apply it [Hickey & Davis, 2004]. This raises issues as lack of formality, analyst dependency and difficulties for less experienced analysts, all added to the profound dialogue gap between business and information systems.

Overall, the requirements of a business process can be classified into functional and nonfunctional, and while traditionally there is a comprehensive coverage on its functional characteristics, the nonfunctional part is treated in a less rigorous way. These later have been associated to the quality factors of business processes, defined and categorized into different dimensions, but with diverse perceptions from several authors and still lacking widely accepted approaches for its criteria and evaluation. So, although an information system's utility is determined by both its functionality and its nonfunctional characteristics, most existing requirements models and specification languages lack a proper treatment of quality characteristics [Chung, Cesar, & Leite, 2009].

Regarding the functional requirements we center our attention on use-cases models, defined inside the Unified Modeling Language (UML) [Jacobson *et al.*, 1999]. Use-cases are one of the most popular techniques for eliciting functional requirements in the design of information systems, whether it involves the development of a new system or the reengineering of business processes.

For nonfunctional requirements we proclaim the use of goals and rules, in their associated combination of checklists and guidelines from Rational Unified Process (RUP) [Kroll & Kruchten, 2003], and in the business plans representation of the Business Motivation Model (BMM)[6]. Notwithstanding other elicitation methods and techniques for eliciting associated goals and rules, as i* or KAOS, this choice is due to the more complete and business oriented side of RUP and BMM. It helps in defining the business requirements specification for business modeling, and promoting the aligning of information systems with business questions that are comprised in process-oriented approaches.

The use and adaptation of "standard" methods and techniques to infer goals and rules requirements from scenarios and process-like diagrams, allows mapping backwardly the traditional business to process workflow. This can allow for better and continuous alignment between business and information systems, and also provide improved traceability, building a strong focus on the business model strategy. This top-down approach initiates on the top use-cases with the elicitation of the respective goals and rules, enabling to drill down into the leaf use-cases by refinement of goals, including objectives, and rules, with its associated strategies, tactics and policies. Accordingly, the knowledge represented in terms of goals, rules and methods can make reengineering tasks more systematic and effective [Yu & Mylopoulos, 1994].

[6] http://www.omg.org/spec/BMM/

2.2.2 Functional Requirements

Inside the RE Object Oriented (RE-OO) approaches, the Unified Modeling Language (UML) contains several techniques with established yet flexible notations and formats such as use-cases diagrams, use-case descriptions, and class diagrams. Use-cases are essentially abstractions of scenarios describing the functional behavior of the system, especially accepted in both research and practice despite short-comings as impreciseness. Diagrammatic and tabular representations make them easy to understand and flexible enough to accommodate some context specific information, being especially effective in projects where there is a high level of uncertainty or when analysts are not experts in the domain [Zowghi & Coulin, 2005].

As the modeling literature has focused on techniques for functional design requirements, with most of these methods introduced as individual techniques for representing an application domain, recent trends have been toward integrating across multiple perspectives. In addition to the widespread use of data models, several organizations denote sophisticated process modeling activity, including widespread application of use-cases as a central aspect of their specification activity. It occurs even in situations where other elements of UML were not fully adopted, and modeling was often described as informal and involving extensive use of natural language narratives [Hansen, Berente, & Lyytinen, 2009].

This significant adoption of use-cases brings greater precision in designer-user communication, but fulfilled through semi-structured natural language exchanges, demanding for formal models with well-organized and structured natural language representations [Henderson-Sellers, Zowghi, Klemola, & Parasuram, 2002]. As many practitioners and researchers have provided guidelines, suggestions and techniques to construct high quality use-case models, systematic reviews are conducted regularly in order to identify and amalgamate that knowledge [El-Attar & Miller, 2012].

2.2.3 Nonfunctional Requirements

Recently, a considerable number of methods and tools have been proposed for the treatment of nonfunctional requirements (NFR), due to the ample evidence that these play a significant role in the information systems engineering process. However, an absence of agreed positions regarding their definition, classification and presentation is still observed. So, further discussion is needed in order to categorize them into their respective positions in the NFR engineering process, as well as their scopes and characteristics to guide system developers in deriving appropriate methods and tools for their treatment [Hasan, Loucopoulos, & Nikolaidou, 2014].

Much work is still required in the systematic process of NFR engineering since all activities are isolated, and there is a disorderly sequence of various methods and tools. Also, more efforts are necessary in order to put forward the design of a comprehensive NFR meta-modeling architecture of sequentially ordered activities with suitable methods and techniques.

Moreover, NFR have been associated to the elicitation of business goals and rules, with important references in the business and information systems domains as the Rational Unified Process (RUP), the Business Motivation Model (BMM) and the Balanced Scorecard (BSC). Nowadays, in order for a software system to be of value, it should meet both functional and nonfunctional requirements, these last by using a goal-oriented representation [Supakkul & Chung, 2005].

Business Goals Elicitation

A recurrent question in research over business goals elicitation is that use-case notation is intended for functional requirements and not nonfunctional requirements, which oversimplifies assumptions on the problem domain. In recent years, goal-oriented requirements engineering (GORE) current states and trends from the viewpoints of both academia and industry have been fully scrutinized, with results pointing for goal models to be useful for supporting the decision making process in the early requirements phase [Yamamoto, Kaiya, Karl, & Bleistein, 2006].

GORE is generally complementary to other approaches, well suited to analyzing requirements early in the software development cycle, especially with respect to nonfunctional requirements. Goal models represent an essential tool for requirements engineers, system architects, and software developers, but their analysis and evaluation also lead to many challenges [Amyot *et al.*, 2010]. A great variety of techniques for analyzing goal models have been proposed in recent years, but, on the other hand, this diversity creates a barrier for widespread adoption of such techniques, also due to the lack of guidance in literature on which one to choose [Horkoff & Yu, 2011].

According to the RUP guidelines, the characteristics of a good business use-case model include the alignment between use-case and strategy, with at least one business goal for each use-case. This is also supported by the BSC classification technique, which allows clarifying the strategic objectives and identifying the critical areas of the organization. Its purpose is to translate the strategy into goals at different levels, and to provide concrete, measurable objectives, which can be directly supported by business processes, ensuring that these are aligned with the business strategy. A business goal is a requirement that must be satisfied by the business and directly connected with the use-case model, the "glue" between business strategy and business use-case.

According to the BMM, the fundamental contribute of goals in the development of a business model is through the definition of Ends and Means of business plans. Among the Ends are the goals and objectives to achieve, which may be either a Vision, an overall image of what the organization wants to be, or Desired Results, more specific intents the organization wants to achieve. These are supported by Courses of Action, Goals supported by Strategies and Objectives achieved by Tactics. A Goal is a statement about a state or condition of the enterprise to be brought about or sustained through appropriate Means (Rules-associated), while an Objective quantifies a Goal, providing the measures to determine whether it is being achieved. An Objective is consistent with the SMART (Specific, Measurable, Attainable, Relevant and Time-based) criteria.

Business Rules Elicitation

Complementarily, business rules are also an important artifact in the requirements elicitation process of IS and a vital part in its development cycle. They describe ongoing policies, procedures and constraints, which concern an organization in order to achieve its business goals and objectives, helping to collect and organize supports for the implementation of change [Kardasis & Loucopoulos, 2005].

In recent years, much research effort has been done in order to unlock this valuable asset that many organizations have concealed [Shao & Pound, 1999]. Its concept has been examined from different points of view, whether as extensions of business goals, or as limitations or constraints on business activities. By structuring, organizing and expressing tactics and policies, close to business viewpoints, it helps collecting and organizing supports for the implementation of change on a business level for the associated IS.

It is important for software to evolve according to changes in its business environment, having business rules as an integral part of the software system, its management and evolution. This improves requirements traceability in design as well as minimizes the efforts of changes, as when requirements are systematically identified and linked to design elements, these are easier to locate and implement [Wan-Kadir & Loucopoulos, 2004].

Even so, the quality of information systems and more specifically of software engineering projects, suffers due to the large gap between the way stakeholders present their requirements and the way analysts capture and express those requirements. One of the problems has to do with the representation of business rules, while another issue, detected in the analysis of current IS requirements elicitation techniques, is that they tend to be much analyst-oriented and dependent [Kapočius & Butleris, 2006].

Regarding the RUP approach, as goals elicitation redirects us to BSC techniques to answer "why" questions, rules elicitation states the "how" vision and the formal *vs* natural language issues. While formalism permits automation, where Object Constraint Language (OCL) is a possibility, natural language stands as the stakeholders preferred technique as it allows for a more direct description and understanding of the business.

By definition, business rules are declarations of policies or conditions, and might apply always (invariants) or under a specific condition, being constructed by constraint (restricting its structure and behavior) or by derivation (inferring facts from other facts). Their purpose is to define specific constraints or invariants that must be satisfied by the business.

According to BMM, business rules are represented through Means, which may be either a Mission, a Course of Action (Strategy or Tactic), or a Directive (Business Policy or Rule items) to employ in achieving Ends. Directives govern Courses of Action and support the achievement of Ends, but they cannot stand on their own, while Courses of Action can be formulated based on Directives, serving as its source, or can be defined to support the achievement of a Desired Result. Strategies are implemented by Tactics and usually channel efforts towards Goals, while Tactics towards Objectives. Business Rules items guide the Business Process and govern its Course of Action. A Business Policy provides the basis for the Business Rules items and governs Business Processes, existing to control, guide and shape the Strategies and Tactics.

2.3 BUSINESS MODELS

The business model (BM) as an instrument for aligning information systems with business is a relatively recent topic of research, one which among other diverse and complementary BM perspectives reflects the reality of an organization [Jensen, 2014]. On the other hand, the BM as a unit of analysis in business practice has a larger research history, having attracted increased interest from managers and researchers. Nevertheless, common consensus is that knowledge on business models is still much unstructured and quite fragmented [DaSilva & Trkman, 2013].

A few years ago, when BM research still traditionally belonged to the agenda of organizational management, the raising debate around it and the lack of theoretical tools in the IS/IT literature, led to a thorough literature review on this topic [Pateli & Giaglis, 2004]. The main concerns for practitioners at that time were the outdated traditional business models and the arrival of new ones that better exploited the opportunities enabled by technology innovations. This added to the increased choices for their design and implementation, turned it into a more complex and difficult task, as also the lack of an underlying common framework of discussion and interpretation of research on business models.

This bottom-up review originated a proposal for an analytic framework categorizing current research and knowledge in the area of BM, examining existing research and classifying it into eight sub-domains: definitions, components, taxonomies, conceptual models, design methods and tools, adoption factors, evaluation models and change methodologies. Complementary, it identified knowledge gaps and proposed future research directions which benefit all. Among them are the building of conceptual frameworks for BM analysis from different viewpoints, the specification of boundaries and identification of each conceptual level, the development of an integrated concept, and the efforts in the area of design methods and tools.

At this time the BM field had matured and was ready to move beyond, with the urgency to create a common language in terms of terminology and conceptualization. In line with this, the existing perspectives were reviewed and an integrated framework for characterizing BM in a six-component entrepreneur-oriented view was proposed [Morris, Schindehutte, & Allen, 2005].

This framework consists of three increasingly specific levels of decision making, considering within each level six basic decision areas with questions that underlie a BM. These decisions represent a hierarchy perspective, progressively moving from the economic to the operational and then to the strategic levels. This permits the user to design, describe, categorize, criticize and analyze a BM, which was defined in terms of a standardized set of quantified decisions, allowing comparisons across models.

The search for new avenues of empirical research, creation of general model taxonomies and investigation of relationships among the foundation level, variables of business modeling, continued with greater intensity since then. This is especially true in the alignment between business and information systems, where the concept can function as a mutual means of communication between both domains. Also, the BM can be used during the development of enterprise applications, as they represent the business logic of a company, and for requirements engineering, by gathering and representing high-level business goals.

One of the most relevant research studies on this topic was the Business Model Ontology (BMO) [Osterwalder, Pigneur, & Tucci, 2005], who's work outlined the general uses, roles and potential of the BM concept in organizations, emphasizing in particular the importance of the concept in relation with information systems. Proposing a terminology or ontology to describe a BM, the BMO works as a reference model shared among the community of practice, defining the terms, concepts and relationships of a BM.

Among its important contributes, we highlight:

- the clarification on the definitions, metamodels, taxonomies of types and instances of the BM concept, and also its evolution (Fig. 2.1);
- the place of the BM concept in the firm, the clear distinction between the terms "business model" and "business process model", as they represent and are used for different

aspects, where business process models are focused on the "how", while business models aim at describing the "what";

- establishment of the Business Triangle, consisting in the constantly challengeable relation between strategy, organization and information system.

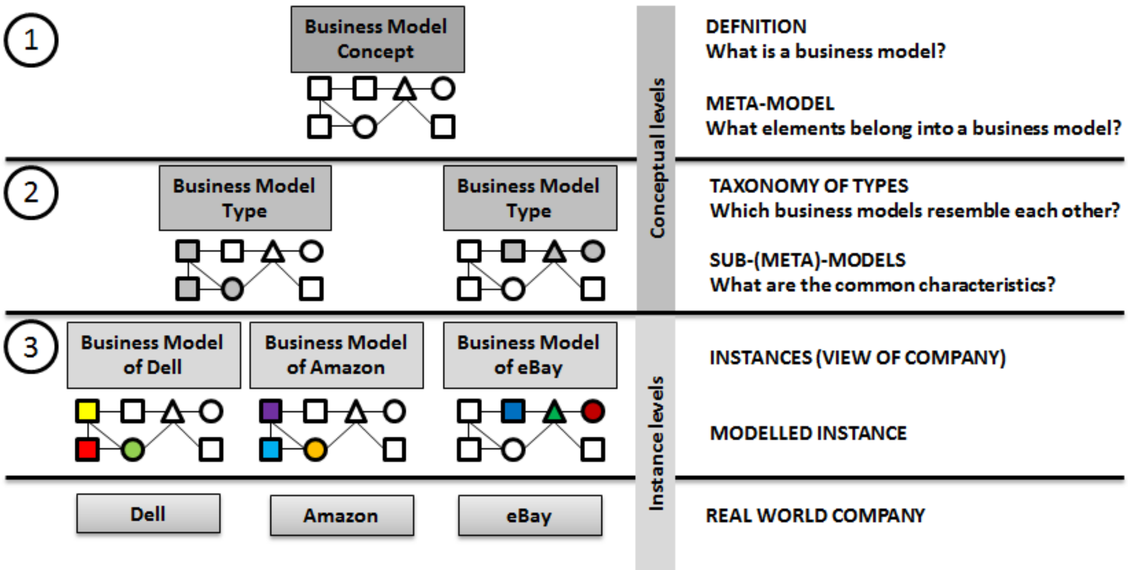


Fig. 2.1: Business Model Concept Hierarchy (adapted from [Osterwalder *et al.*, 2005])

Relatively to the BM impact in information systems, a clear concept would contribute to the creation of a common understanding between business and information systems, leading to a strategic and functional integration, and a more efficient IS/IT infrastructure. Also, it could become the conceptual tool to capture, share and create a common vision of a company's BM, illustrating its business strategy and objectives, while integrating it with the enterprise and information system models (Fig. 2.2).

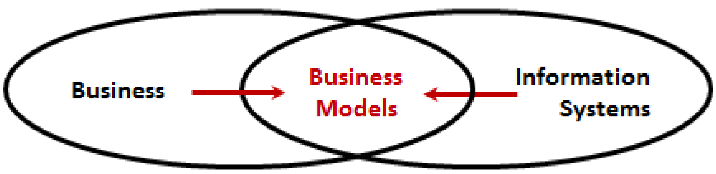


Fig. 2.2: Business Strategy and IS Alignment (adapted from [Osterwalder *et al.*, 2005])

Clarification of the concept and its orientation for information systems would also be important to:

- develop conceptual approaches for designing new computer-based BM tools;
- contribute to requirements engineering, particularly in the process of defining business goals and finding innovative ways to model business requirements and improve the aligning of information systems with business;
- capturing, mapping and following the BM of an organization as a form of knowledge management;
- establish the BM role in defining goals, workflows and processes.

Trying to reach a consensus and overcome one of the main deadlocks in the topic, a BM definition, still very popular these days, was proposed:

> *"A business model is a conceptual tool that contains a set of elements and their relationships and allows expressing the business logic of a specific firm. It is a description of the value a company offers to one or several segments of customers and of the architecture of the firm and its network of partners for creating, marketing, and delivering this value and relationship capital, to generate profitable and sustainable revenue streams."* [Osterwalder *et al.*, 2005]

Another central contribution of the BMO work is the domain synthesis, consisting in nine building blocks that cover all the BM components mentioned by at least two authors in literature. The continued dedication to this topic led to the popular Business Model Canvas (BMC) [Osterwalder & Pigneur, 2010]. The BMC works as a strategic management template for developing new or documenting existing BM, consisting in a visual chart with elements describing an organization's value proposition and its infrastructure, customers and finances.

Another interesting research using ontologies [Andersson *et al.*, 2006], proposed a reference ontology of BM using concepts from three already established BM ontologies (REA, e^3-value and BMO). By performing a thorough analysis, the authors introduced a number of additional concepts and increased the understanding of the original ontologies, complementing and improving them. Their work has also shown that there is a considerable overlap between the ontologies. Although more situated in the managerial domain and not yet validated in practice, the results of the ontology from this research may serve as a mapping tool, where models can be transformed from one formalism to another.

Attesting the growing importance and the need to focus research on the concept of BM, a schema for BM research in a structured manner, from conceptual to theoretical, drawing on the scientific and business research literature, was suggested [Lambert, 2006]. Using a combination of both inductive and deductive research methods, it promoted the creation of taxonomies and typologies, respectively, in order to advance the knowledge in the area. Nevertheless, developing a general taxonomy of BM is only possible when a widely agreed-upon concept of a BM exists and current deductive research efforts turn towards BM theories.

Following the rising importance of the BM concept in the field of information systems research, while answering this call for a general taxonomy, a deductive reasoning approach was used to organize a hierarchical taxonomy on the associated BM concepts. Accordingly, a comprehensive framework was developed, forming a complete ontological structure of the BM concept [Al-Debei & Avison, 2010]. Through this work, four primary BM dimensions along with their constituent elements were identified and the BM modeling principles were cohesively organized. This explained the reach of the concept and showed its interaction and intersection with strategy, business processes and information systems.

This advance also renewed the calling for a consensus on BM concept and composition, as it is crucial for researchers to have a reference measure on which to apply in different industries and within different contexts. By setting the main ontological dimensions as value proposition, network, architecture and finance, while stating the BM as an essential, conceptual, multi-level, dynamic, granular and coherent tool of alignment in digital business, the urge persisted for the need to review continually any BM. This is necessary in order to ensure its fit with the complex, uncertain and rapidly changing external environment. Special attention is needed as this change comes also from inside the business strategy, business processes and information systems, which are all connected, in order to build and maintain the most appropriate BM to achieve the strategic goals and objectives.

As in many situations throughout this research topic, no common consensus regarding the delimitation between BM and strategy can be reached. These two concepts are very close to each other, as BM is not a strategy but includes a number of strategy elements, nonetheless, in the end, BM are a reflection of the organization's realized strategy. Following works supported this conclusion, as in the case of a study over the notions of strategy, BM and tactics [Casadesus-Masanell & Ricart, 2010]. Although in simple competitive situations there is a one-to-one mapping between strategy and BM, these concepts are related but different. Moreover, sustaining that in the managerial domain designing a new BM is closer to an art than a science the call for clear definitions of constructs in future research remains.

In a recent reference in this area, following on the conclusions of an early work [Pateli & Giaglis, 2004], an up-to-date on the literature analysis was performed to observe changes and review existing perspectives since then, proposing and using a comparative research framework [Burkhart, Krumeich, Werth, & Loos, 2011]. Results of the analysis showed many research gaps still existent in the field of BM research, in spite of all the calls for clarification and synthesizing of the concept of BM, which have to be overcome to achieve a successfully usage. Despite the many scientific papers focusing on defining a BM, no generally accepted definition was found so far and progress in the field has been hindered by lack of consensus over its key components.

Many points are in favor of BM, with multi and cross-disciplinary view on BM as an open field of research where its concept can be employed for the alignment between business and information systems, functioning as a mutual means of communication between both domains. Accordingly, it can be utilized during the development of enterprise applications, where the concept can be used for requirements engineering by gathering and representing high-level business goals. On the other hand many challenges remain, as the limited knowledge on influencing factors and interdependencies of BM components, its uncertain representation as there is no commonly accepted notation, and the lack of evaluation criteria and corresponding metrics to assess them.

2.3.1 Business Model Canvas

The Business Model Canvas (BMC), a strategic management template for developing new or documenting existing business models, currently stands out as one of the preferred tools for their generation, especially in business related audiences. The BMC is based on the Business Model Ontology (BMO) [Osterwalder, 2004], where the formal descriptions of the business become the building blocks for its activities. These are divided in nine different business conceptualizations, organized by four dimensions: Infrastructure, Offering, Customers and Finance. In turn, this division was based on the early work of the Balanced Scorecard (BSC) four perspectives: Financial, Customer, Internal Business Process and Learning & Growth [Kaplan & Norton, 1996].

Regarding BMC, the value proposition describes which customers' problems are solved and why the offer is more valuable than the competitors' similar ones. Customers themselves are analyzed in segments and relationships, while the distribution channels means to illustrate how the customer needs to be reached and by whom. Resources are transformed through the key activities into the final product or service, which depends on external partner networks. Financial information focus on cost and revenue, where cost structures should be aligned to the core ideas and revenue streams must mirror the value assigned by customers, in terms of how much they are willing to pay and how they will perform transactions.

Accordingly, BSC management perspectives track its four high-level types of measures, using this approach to identify corporate objectives within each of the categories and seeking to align the balanced set of measures with the organization strategy. It uses a top-down method that defines the organization strategic goals, passing these goals downward, whose success results form a strategy-focused organization, derived from strategy maps and balanced scorecards. A strategy map shows how the organization creates value and describes how organization key business objectives align within the four BSC perspectives to support corporate strategies.

BMC and BSC are two different but complementary tools to achieve innovation, tactical directions and action plans in an existing or planned organization. While BMC determines part of the business strategy, BSC is aimed to track implementation and ensure that the organization strategy is executed. Recent research classified BMC and twenty nine other relevant literature sources on business models [Burkhart *et al.*, 2011], with BMC obtaining interesting global results: positive on 66,7% of all the criteria analyzed, and checked on all of the top six criteria items and on 50% (six out of twelve) of the second-level ones.

2.3.2 Balanced Scorecard

The BSC stands as a reference in the strategy definition and orientation of an organization, being closely related to its business model. Its use has been increasing, within varying degrees of adaptation, as a number of organizations are actively using it to link their strategy and operations. Also, the contribution to an organization's strategy implementation, subject to the adoption of suitable processes, can provide a mechanism for addressing key problems by making explicit the link between strategic objectives and operational goals. Nevertheless, there is no warranty whether BSC can effectively enable strategy implementation by itself or in determining how effective it is in its strategic initiative [Atkinson, 2006].

Much research has been focused on crossing or complementing BSC with other methods, techniques and frameworks, as a way to overcome criticisms and some of its limitations. Examples as the use of scenario planning to cover the lack of external orientation in BSC, helping to make it more reflective of changes that may appear in the future and ensuring that the system implemented by the organization is linked to the realities in the environment [Othman, 2008].

On another example, is the comparison of the BSC framework with the Goal-Question-Metric (GQM) technique, a quantitative approach for developing and supporting software process improvements, where the added value of BSC resides in the causal relationship chain among the business goals of the various perspectives, enabling a proper alignment of business and operative goals for achieving success [Buglione & Abran, 2000].

Furthermore, the combination of BSC and data envelopment analysis (DEA) is another proposed solution. By using BSC as a comprehensive framework for defining information systems projects evaluation criteria and DEA as a nonparametric technique for ranking those projects, they are integrated into a new approach [Asosheh, Nalchigar, & Jamporazmey, 2010].

Several studies on BSC analyzed the strengths and weaknesses of this management tool [Self, 2004], as within its limitations, it provides a snapshot of organizational health, but it does not give a three-dimensional picture, it can point out problems, but it does not reveal the solution. A significant number of one-dimensional models of performance already exist, with a tendency to integrate many individual measurements into a single performance index. This can make the overall picture somewhat fuzzy, while adding significantly to the workload [Buglione & Abran, 2002].

The question is that one-dimensional models do not meet the analytical requirements of management when various "viewpoints" must be taken into account simultaneously. One solution handling a greater number of perspectives, integrating the BSC and the QEST nD model [Abran & Buglione, 2003], has shown to be valuable and valid for capturing different measurement viewpoints. Thus, this makes it possible to read data and analyze them with a "layer" logic, from top to bottom, considering each BSC perspective as a separate layer contributing to the final result.

The BSC has been a widely used approach for multidimensional performance measurement in the context of information systems management, though the strong decrease in its application between 2006 and 2009. Nevertheless, recent publications indicate a renewed increasing attention to its usage in information systems [Gyory, Brenner, & Uebernickel, 2012]. As information systems services are becoming increasingly critical for daily operations, demanding more transparency and control, there are however few available tools to evaluate the information system service portfolio. This applies accordingly to the strategic goals of a company from the business perspective, or to support business-oriented service and service portfolio decision-making, in order to achieve alignment between business and information systems.

Following this trend, a multiple perspectives goal-oriented requirements elicitation framework was proposed [Lee-Klenz, Sampaio, & Wood-Harper, 2010]. It is aimed at identifying information systems service requirements, while supported by a tool to help requirements analysts attain the alignment of business goals with services. This framework is based in the adoption of the goal concept from KAOS, extending it towards capturing multiple perspectives in early requirements, while integrating the BSC and software measurements to information system goals, strongly based on bringing socio-technical methods to information systems analysis. Used in a number of successful industrial projects, it had a positive impact in eliciting nontechnical information relating to strategic and operational intent, which are seldom addressed by traditional software development approaches.

2.4 ENTERPRISE ARCHITECTURES

Enterprise Architecture (EA) research had its first major impulse in the late 1980's [Zachman, 1987] and in the early 1990's [Sowa & Zachman, 1992] with the Zachman Framework, which still remains today as an inevitable reference in the field. This work presented a framework to tackle the increasing size and complexity of the implementation of information systems and the difficulties of communicating with one another about information systems architectures.

The first work provides a taxonomy for relating the concepts that describe the real world to the concepts that describe an information systems architecture and its implementation, while in the second, where the "enterprise" term is used extensively, the framework is extended and formalized conceptually. Though all its merits, the Zachman Framework has some drawbacks, as it lacks coherent and consistent models for its cells [Ostadzadeh, 2007], and a methodology with a well-defined repository and a popular modeling notation [Fatolahi & Shams, 2006].

The topic of aligning information systems to EA was tackled early on [Brown & Magill, 1994], with attempts to identify the best way to organize the information systems functions within an enterprise, a critical information systems management issue at the time. Topics on the vision, strategy and structure of the overall organization, and the information systems infrastructure and its increasing strategic role shaped the frameworks for analyzing the information systems alignment decisions.

However, conceiving, planning and monitoring these systems is not trivial, and while it sometimes may seem like mission impossible, it is in fact mission critical. With time, creating a framework for an enterprise could be as simple as tweaking an existing framework or as complicated as inventing its own, always by carefully evaluating and understanding each business environment [Armour, Kaisler, & Liu, 1999].

The scattering of frameworks with no methods, or with methods and tools that gave very little guidance with regard to how and why to use them in different situations, reinforced the need for the modeling topic in the field of EA. Nevertheless, many researchers and practitioners still model just for the sake of modeling, neglecting the practical use of methods and related tools for development of the business. Also, they disregard to safeguard its quality for creating visions and strategies, redesigning the business and developing information systems, while encompassing all related sub-models of the enterprise perspectives [Persson & Stirna, 2001].

The proliferation of enterprise modeling languages and tools, and the emergence of workflow management systems and their myriad of commercial tools, all of which were not interoperable and could not exchange models, led to a tower of Babel situation [Vernadat, 2002]. On the basis for the development of a unified and universal language for enterprise modeling, the Unified Enterprise Modeling Language (UEML) was proposed with the aim of supporting integrated use of enterprise and information systems models. Based on UEML, another interesting language was proposed, the Unified Enterprise Competence Modeling Language (UECML), with the objective of modeling an enterprise in terms of competences and capabilities, a standardized way to define the profile of an enterprise [Pépiot, Cheikhrouhou, Furbringer, & Glardon, 2007].

Another relevant approach in enterprise modeling languages is the Multi-Perspective Enterprise Modeling (MEMO) [Frank, 2002]. It is based on an extendable set of special purpose modeling languages for describing corporate strategies, business processes, resources or information. One of its strongest points is that its languages are defined in metamodels, which in turn are specified through a common meta-metamodel, which allows providing intuitive abstractions for various observers, making available different views and perspectives on an enterprise.

Also, in a correspondent work and based on some similar concepts, an extension of an approach to enterprise architecture and its implementation using a commercial metamodeling platform was proposed [Braun & Winter, 2005]. This solution allows analyzing the strategy, organization and system layers and its intra and inter-layer consistency issues, and reinstating the need to design the information system and its processes according to its business requirements. In doing so, it identifies that not all layers can be found in all existing approaches. Again, the research community called for a metamodel that is capable of representing all relevant artifacts of enterprise architecture consistently, both from business and information system perspectives. Also, it should be implemented using an appropriate modeling tool, in order to support efficient and computer-based architecture management processes.

On a different approach, originally intended towards generating a process support environment, the Extended Enterprise Modeling Language (EEML) was used for a more general enterprise modeling [Krogstie, 2008], extending it to support modeling of formal rules. In the process of developing a rule-based system for rule automation, supporting process-modeling and goal-oriented modeling as part of requirements specification, the language was even evaluated relative to emergent standards as the OMG's Semantics of Business Vocabularies and Rules (SBVR)[7].

Recent work on the UEML framework refreshes and tries to motivate researchers for the existence of the UEML approach, and also points out its benefits and paths for further work [Anaya *et al.*, 2010]. As metamodels and ontologies are another current trend, paths for further work could be to connect the meta-metamodels for language and construct description, and for its common ontology to connect with the quality framework. From all the languages revised, UEML is a promising, ambitious, long-term project, which may even be useful outside enterprise and information system modeling, one that is worth being aware of in the future.

Recent studies in literature for the most commonly cited benefits on the use of EA, have the alignment between business and information systems appearing consistently in the top-5 [Tamm, Seddon, Shanks, & Reynolds, 2011]. The reasons for this are that the business analysis undertaken during EA planning spans different departments and business units of the organization, as well as its sub-dimension of alignment between business and information systems, leading to an improved performance. Also recently, the aligning of information systems with business, in the scope of organizational alignment, has been receiving extensive attention in the information systems literature, with a positive impact in several dimensions.

Moreover, practitioners seem to be following EA research trends, reaching for ever-higher maturity levels. Reports on using individual customized approaches of EA frameworks, by leveraging elements of the available general ones, point to The Open Group Architecture Framework (TOGAF)[8], Architecture of Integrated Information Systems (ARIS)[9] and Zachman as the most popular references in the field [Lange & Mendling, 2011]. ARIS is a popular approach to enterprise modeling by the group Software AG, and TOGAF is a framework which provides a comprehensive approach for designing, planning, implementing and governing enterprise information architectures.

2.4.1 Enterprise Modeling

Regarding alignment in the field of Enterprise Modeling (EM), only recently the research on EA evolved substantially. Supported on the basis of an integrated and coherent description of EA that provides insight, enables communication among stakeholders and guides complicated change processes, a thorough proposal was outlined: ARCHIMATE [Jonkers, Lankhorst, & Buuren, 2004]. This language for EA description allows architects to use their own modeling techniques and concepts within each architectural domain. Its concepts establish the relationships between architecture descriptions at the business, application and technology levels, playing a central role related to the ubiquitous problem of the alignment between business and information systems.

[7] http://www.omg.org/spec/SBVR/

[8] https://www.opengroup.org/togaf/

[9] http://en.wikipedia.org/wiki/Architecture_of_Integrated_Information_Systems

ARCHIMATE was compared to a selection of standards and languages, finding limitations in each of them, and proposing to overcome those limitations. The concepts of this language propose to hold the middle between the detailed concepts that are used for modeling individual domains and very general architecture concepts that view systems merely as entities and their inter-relations. It forms a basis for bridging the heterogeneity of existing languages, while also claiming to be the only EA description language that fully enables integrated EM, with a designed integration tool workbench for it [Lankhorst, 2004].

The developed workbench allows the concurrent design of enterprise architecture domains, where each domain can still be designed using its own languages, tools and techniques. Also, as it integrates all languages, it has the ability to communicate and reason across domain boundaries, and so the workbench becomes an instrument for collaborative design of enterprise architectures. Another important functionality of the workbench architecture is its feasibility for transformations between any tool-specific and ARCHIMATE contents.

Goal Modeling

ARCHIMATE and most EA modeling approaches deal with structure, function and behavior, neglecting the intentional dimension of motivations, rationales and goals. Though these are important, long recognized, elements in the field, exploring the potentials of intentional modeling in EA, it is also vital to analyze the prospects and challenges of incorporating intentional modeling concepts in the EA construction process. One example is, namely, by assessing related languages such as OMG's BMM and i* [Yu, Strohmaier, & Deng, 2006].

As the motivation knowledge is poorly supported, which is typically embedded in documents, meeting minutes or held in the minds of individuals involved, the context, necessity and justifying of past actions, or revisiting decisions, cannot be traced back. The solution to these and other related problems requires the "why" knowledge to be organized and linked to the "how", so new kinds of analysis can be performed. This latest proposal uses BMM to capture the high-level business motivations and then i* to further refine and analyze them, using the semantic of intentional concepts. In this stance, BMM organizes the "why" knowledge and i* links the "why" knowledge to the "how". Also, referenced elements in BMM can be related to each other and further refined through i* until any level of detail.

Subsequently, to solve this insufficiency in ARCHIMATE a language based on existing work on high-level goal and requirements modeling, ARMOR, was proposed in order to support the modeling of the enterprise motivation, extending the ARCHIMATE modeling framework with a fourth aspect [Quartel, Engelsman, Jonkers, & Sinderen, 2009]. The ARMOR language was designed to model the motivation of EA, namely goals and requirements, and illustrate its potential use for analyzing enterprise architectures, while maintaining focus on the alignment between business and information systems issues. The extension consists in motivation and meaning aspects, concerned with the goals and intentions of the enterprise, while representing concerns that are related to the semantics of EA artifacts respectively, and a value layer, representing the value of services and products that are offered to customers.

Added to its alignment to ARCHIMATE, the ARMOR language intends to align with languages for requirements modeling, namely the BMM, the i* framework and the KAOS notation, reusing concepts from all of them. Also, it aims to enable documentation, communication and reasoning about requirements, providing ease of use, extensibility and traceability. Although BMM cannot be strictly considered a true requirements modeling language, a business plan can be used as a starting point for the early requirements engineering phase. The i* framework operates on the same phase but with a more expressive language, allowing various types of analysis. As opposed, the KAOS graphical notation is less expressive but easier to use than i*.

The key concept used to align the conceptual model of ARMOR with BMM, i* and KAOS is the concept of goal, which is supported by all of them, as also is the refinement of goals into sub-goals. Here, a goal is defined as some desired effect in the problem domain or some desired properties of a solution. On the other hand, unlike BMM, both i* and KAOS allow modeling goals or situations that may have a negative influence on the satisfaction of another goal. Additionally, modular extensions of ARMOR to support modeling the stakeholders' domain were presented, as well as the use-cases and business rules, and metamodels for all of them.

Alongside and following the goal domain research topic, a review was performed on its support in several EM approaches and EA frameworks, particularly in the way in which each of them proposes to align goals with the remaining elements of the enterprise [Cardoso, Almeida, & Guizzardi, 2009]. The motivation for this review was the little attention devoted by the research community to the association between goal models and other elements of the EA in the context of EM. Again, goal is the key concept, with goal modeling being the artifact employed for capturing the motivational aspect and strategies behind the organizational practices, documenting it, and helping in clarifying interests and intentions from different stakeholders.

Also, goals are very important as they affect other architectural domains by providing a strategic dimension for business processes, resources, organizational actors and roles. The observations over current approaches ranged from the rudimentary notion of intentional aspects provided by ARIS, to the more sophisticated support provided by the ARMOR language. With ARCHIMATE already established as a The Open Group standard, a refined and extended version of ARMOR was presented [Engelsman *et al.*, 2011], adding concepts also used in TOGAF's architecture development method (ADM), which acknowledges goals and requirements as central drivers in the architecture development process.

Although KISS (keep it small and simple) was one of the guidelines on ARMOR development, there were some usability and utility difficulties working with enterprise architects who already understood and used ARCHIMATE, as the language was too complex for some of them to understand [Engelsman & Wieringa, 2011]. After redefining ARMOR to contain only a minimum number of goal-oriented concepts, it was still found useful for traceability management in practice. A stripped down version of the ARMOR language, called Light ARMOR, dropped some constructs, extended or replaced others, and left requirements as simply goals, with no separate concept for them.

2.4.2 Architecture and Business Model

On a different direction, a recent work referencing ARCHIMATE presented a model construct similar to an enterprise architecture model, but with the addition of a strong business model component to allow for a better alignment, the Business Model Ontology (BMO) [Fritscher & Pigneur, 2011]. A visualization comparing information systems services and business model was proposed, providing a more business oriented view on enterprise architecture, which has models for business, application and infrastructure levels. This solution facilitates alignment between the strategic business vision and the strategic information system considerations.

The mixing of the ARCHIMATE information systems architecture and the BMO business model allows perceiving the connections, helps in aligning and highlighting the interactions between them, and opens the possibility to have both a top-down and a bottom-up, business and information systems focused view of the enterprise architecture. Additionally, it is intended that this solution can identify misalignments, deploy a more systematic way of transition between both models, and have the ability to highlight and compare business model patterns.

In the same line of relating business model to enterprise architecture, again using BMO and ARCHIMATE, a mapping between the two associated metamodels was proposed [Meertens *et al.*, 2012]. BMO had no official or complete metamodel, so there was a need to create one and also to develop two extensions to ARCHIMATE. The alignment of the de facto standards in their respective fields allowed going from business definition and strategy, to business goals and into technology infrastructure. This facilitated the traceability of business requirements, defining and analyzing business alternatives, and assessing the impact of a change in the architecture for an existing business model.

Findings are that any BMO concept can be mapped to ARCHIMATE, but the reverse is not always possible due to its richer language. Even reducing this complexity, taking into account only the business layer of the original metamodel of ARCHIMATE, there is an excess between the two regarding the concepts event, meaning and representation. In the opposite direction, there is the situation of overload, with different BMO concepts mapping onto the same ARCHIMATE concept (usual situation in cases of reverse engineering). Also, the possibility of redundancy is ever present, as ARCHIMATE has more than one concept suitable to represent some BMC concepts.

Contrasting with the previous referred work [Fritscher & Pigneur, 2011], the ARCHIMATE team motivation was "to analyze the possibility to extract the business model from the architecture model, by leaving out application, technology details and even business process details". The solution was extended by elaborating on cost analysis techniques having as result more accurate and realistic calculations, and on methodological support complementing the TOGAF ADM [Iacob *et al.*, 2014]. This supplement to previous research clarified the role of business models for business requirements management and for architecture change.

2.4.3 Services on Business and Information Systems

Service support can help in reducing the gap between business and information systems, ensuring a good connection and mapping between models of both sides, thus easing the communication and understanding between them. As the concepts of Service-Oriented Architectures (SOA) apply both to business and system architectures, they can help the business and system stakeholders to align their business requirements and information systems architecture implementations.

Attesting this, recent works ([Delgado, Guzmán, & Piattini, 2010], [Elvesaeter & Panfilenko, 2010]) provide solutions for the service oriented development of business processes, relying on the SOA paradigm closeness to business models while integrating concepts, methodologies and tools that combine the application of Service-Oriented Computing (SOC) and Model-Driven Development (MDD) paradigms to support Business Process Management (BPM). These last through the use of OMG "standards" to model business processes and services, such as BPMN[10] and SoaML[11].

[10] http://www.omg.org/spec/BPMN/

Applying SOC and BPM paradigms in conjunction is an important but no trivial step to take, involving different visions of business and technological challenges, and demanding a service oriented methodology for the derivation of software services from business process models [Delgado, Ruiz, Guzmán, & Piattini, 2009].

Moreover, when in a collaborative context, the integration of different partners deeply depends on the ability to use a collaborative architecture to interact efficiently, and to help bridge the requirements gap between the business analyst and the information systems analyst developer levels. Using business models to design a logical model of a solution is an essential step to reach a collaborative solution and support practices in the development of systems integration [Touzi, Benaben, Pingaud, & Lorré, 2009].

A model-driven approach for the direct generation of services from business processes provides several advantages, such as the reuse of the knowledge embedded in the correspondences defined between the involved metamodels and the traceability between their elements. These advantages allow for a better alignment between business and information systems. Also, it is important to define how people collaborate within these organizations, and how they aim and reach for the same goals.

The realization of business processes by means of services becomes more important, since it is common for participants to collaborate within an organization. In this context, transformations to generate SoaML service models from use-cases models are required in order to obtain service-oriented systems to realize business processes, supporting the separation of their definition from the technologies involved. Recently, a proposal to define a framework for the Model-Driven specification of Software Architectures, which uses the concepts behind Service-Orientation for its definition was presented [Delgado & Ruiz, 2012].

By describing business entities and relationships required by a system software solution and its low-level representation, in which the technological aspects determine the final shape of the system, another solution to the problem of architecting the existing gap between the high-level configuration of an IS, here in the form of a software system, has been proposed [López-Sanz & Marcos, 2012]. This adds to the growing incidence of research work between the service-oriented and software engineering communities, although few of the existing proposals take advantage of the benefits that the use of services offer to the specification of architectures in combination with a model-driven approach.

Most of the proposals providing an architectural approach to Service-Orientation are related to the definition of reference architectures, while on the other hand it is quite rare to find solid proposals that tackle architecture specification from a model-driven perspective that use those principles. Accordingly, the definition of a framework for the model-driven specification of software architectures, using concepts behind service-orientation for its definition, showed a tendency towards the use of a hybrid approach as the most suitable choice, following a 'pure' UML-based approach for architecture specification where SoaML is worth mentioning.

[11] http://www.omg.org/spec/SoaML/

2.5 CONCLUSION

Requirements engineering, as the fundamental, transversal topic that connects all perspectives, together with the business models, a still fuzzy topic but with already solid references available, and the enterprise architectures, focusing on the enterprise modeling, goals, strategy and services related issues, form the basis for the state of the art of the perspectives on aligning information systems with business.

Regarding requirements engineering, it covers more specifically its activity of elicitation for functional and nonfunctional requirements. The focus is on use-cases for the business processes in the functional side and on business goals and rules in the nonfunctional one, both inspired in their handling by the RUP methodology and their representation by the BMM specification.

The evermore growing influence of business models towards information systems is observed, stating the great efforts still to be accomplished, as there is still no consensus on a BM definition. From the several existing proposals two stand out from the crowd, the BMC and the BSC, different but complementary tools for business modeling in organizations. This focus on BMC is due to its current research impact and leadership in professional use, while on BSC it is by its recent revitalization in the information systems research area and also due to its background as the original base support of the BMC itself.

Enterprise architectures have a somewhat long history of frameworks and associated modeling languages and tools, with both commercial and research-oriented proposals. Regarding enterprise modeling, the most popular is ARCHIMATE, counting with motivational goal complements, business model ontological comparisons and connections to service support.

All these three perspectives are highly model-oriented, counting on the support of methodologies and metamodels, as also in the adoption of methods and associated tools. The topic of modeling in the business and information systems areas is extensive, nevertheless some references are essential to position ourselves and set the basis for the work ahead.

—— This page intentionally left blank ——

CHAPTER 3

ADOPTION OF MODELS IN INFORMATION SYSTEMS DEVELOPMENT

Summary: The crucial importance of modeling in information systems development, its widespread use and the need for a grounded set of terms in using it, are discussed across this chapter. Also argued are the benefits on the use of standards alongside its associated metamodels, namely the ones related to the OMG specifications. Moreover, as tools are essential for users to fully profit from the use of modeling and methodologies, especially within methods' tool support, recent trends in prototype development are revised.

CHAPTER TABLE OF CONTENTS

—— This page intentionally left blank ——

CHAPTER 3

ADOPTION OF MODELS IN INFORMATION SYSTEMS DEVELOPMENT

"He who has not first laid his foundations may be able with great ability to lay them afterwards, but they will be laid with trouble to the architect and danger to the building."

– Niccolo Machiavelli, **The Prince** (1532)

3.1 INTRODUCTION

Modeling has always been at the core of both organizational design and IS development, but choosing the right model involves knowledge of the purpose, analysis and availability of the various modeling techniques and tools. In information systems and software development, frameworks, methodologies, methods and modeling, all have their importance; nevertheless we should be cautious with our terms. In this chapter, we include the topics related to the more practical part of our research work, in what involves its associated methodologies, models and tools.

As an intrinsic part of the methodologies that we will use, the OMG Specifications related to the business modeling and management catalog, and a few others more, are reviewed and related with each other. Based on our background formation on mathematics and computer sciences, standardized methods and modeling are an essential part of our work and vision, so the theme and aim of this planning is purposely, and whenever possible, directed to OMG specifications and the development of meta-methods and tools.

Specifically regarding the OMG Specifications, a particular one is related to the business modeling and management catalog, the Business Motivation Model (BMM), and two others to the modeling and metadata catalog, the Service oriented architecture Modeling Language (SoaML) and the Software Process Engineering Metamodel (SPEM). Added to these, the SoaML-related Software Quality Characteristics (CISQ) is also in our range of interests.

A solid business model, describing the business strategy, business vocabulary and business rules, as well as business processes, plays an essential role when developing an IS. The BMM specification standardizes such business models, while SoaML is an interesting solution for a larger, enterprise description of an IS, complementing the business oriented aggregated structure of BMM and relating with CISQ for a more technology-oriented linkage. Furthermore, the SPEM is oriented to the development process cycle, where it can model and integrate with the methodologies and tools presented ahead.

Models, as useful tools for abstraction, need themselves tools to maximize their benefits. This following and also in line with the decision to follow the OMG specifications, namely the UML reference, we look upon UML-based modeling tools and other reference tools, both in practice and in academic environments. Also, in search for a more lightweight approach, one adapted to the context and timespan available for this work, we revise the recent trends in prototype development environments.

3.2 MODELING AND METHODOLOGIES

Despite the importance of modeling, its widespread use raises several problems, with different meanings being associated to it either in business or in information systems. In order to tackle the present chaos in the business process modeling (BPM) and information system modeling (ISM), tentative evaluation frameworks and taxonomies of BPM and ISM techniques have been proposed [Giaglis, 2001]. Their aim was to answer the need to develop a single holistic technique that could effectively represent all modeling perspectives in a rigorous and concise fashion to be applicable in all modeling situations. This could be done by analyzing the extent to which they could provide the ability to model different business and system perspectives.

Also, as the number of references is very high, it is hard to get an overview and understanding on many of the concepts and vocabulary involved. In the case of business modeling, a framework to help in deciding which, among the available techniques, is the most appropriate technique for each case was proposed [Aguilar-Savén, 2004]. This framework tried to answer the need to clarify, classify, organize and structure this field of research, and though it was centered in BPM, its conclusions applied to modeling in IS in general. The fact is that to study and understand systems, people construct models according to particular viewpoints and using particular modeling techniques and tools, but these cannot provide the solution for themselves, they are an aid in designing them, supporting communication and enhancing understanding.

Even though we advocate the use of models and the importance of structured and formal methods and techniques, we are aware that the majority of users do not follow these ideas. This is especially true in practice, as a study about what are the most popular techniques and tools used by analysts when they model, and what are the major purposes for which conceptual modeling is used, shows [Davies, Green, Rosemann, Indulska, & Gallo, 2006]. Not surprisingly, in this study the top most frequently used modeling techniques and methods were Entity-Relationship diagramming, data flow diagramming and systems flowcharting, with workflow modeling and UML with a somewhat disappointing classification. Moreover, modeling technique use was found to decrease significantly from smaller to medium-sized organizations, although it increased significantly in larger organizations.

Consequently, in order to plan, create, test and deploy a solution for an information system, namely by building a range of hardware and software configurations, for each one of them or a combination of the two, a methodology is also needed. Its aim is to produce high quality systems that meet or exceed customer expectations, based on their requirements, by delivering systems which move through each clearly defined phase, within scheduled time-frames and cost estimates. As systems grow in complexity and the everyday practice of information systems development is very diverse, a research framework for its methods and tools is needed, whereas method engineering covers the research field of the construction of information systems development methods (ISDM) and tools [Brinkkemper, 1996].

A myriad of system development models or methodologies have been proposed in literature, divided among paradigms, approaches, methodologies and techniques, raising the need for a framework to classify them according to each foundational features [Iivari, Hirschheim, & Klein, 2001]. The clear definition of major terms as method, technique, tool, and methodology, among others, making sense of the "methodology jungle" derived from the continuous proliferation of new information systems advances, asks for the necessary flexibility for handling, accommodating and assimilating these in a clear and integrated framework. These terms are used extensively in literature, though much of the times inconsistently, which adds to some confusion and ill-definitions both in research and practice.

The early computer applications of the 1960s and 1970s were developed and implemented without explicit or formalized development methodologies, where user needs were rarely well defined, in so originating system designs often inappropriate for meeting genuine user and business requirements. These limitations led to a new appreciation of standards and a more disciplined approach to developing information systems in organizations [Fitzgerald & Avison, 2003], building computer-based applications focused on the identification of phases and stages, namely the Systems Development Life Cycle (SDLC), or, more commonly, the waterfall model.

SDLC was associated with a set of techniques (such as flowcharting) for use in particular phases and the notion of iteration around the phases, an approach which also involved serious limitations. Along the way a number of newer approaches emerged in response to one or more of these limitations, thus launching the methodology era. A methodology is a recommended collection of phases, procedures, rules, techniques, tools, documentation, management and training used to develop a system. These can and should be adapted using other techniques and tools, providing an important contribute in practice, and also on training and education, teaching good practice and forming a solid basis for understanding systems development.

Like most previous methods, their development and promotion have been almost entirely driven by practitioners and consultants, with little participation from the research community during the early stages of evolution, as the low number of conceptual studies on information systems development (ISD) agility in existence shows [Conboy, 2009]. For around the last 10 years or so, there was the emergence of a number of ISD methods, which have collectively been labeled as agile, these again almost completely due to the efforts of practitioners and consultants. The low research effort focused on the conceptual development of agility in the ISD field impacts on agile method knowledge, which now suffers from a number of conceptual problems due to the lack of foundations, presenting significant problems for practice.

The articulation of the agile manifesto in 2001 has brought unprecedented changes to the software engineering field [Dingsøyr, Nerur, Balijepally, & Moe, 2012], witnessing the introduction of many software methods, tools, techniques and best practices readily accepted by many practitioners. Nevertheless, much work has still to be undertaken to bring coherence to the current discourse on agility, in order to consolidate the move towards collaborative development. This leads to people being accorded privileges over processes that formerly constrained them, and in minimizing unnecessary work, particularly with regard to the creation of wasteful documentation.

This move allowed customers and stakeholders to actively shape and guide the evolution of the end system, with an acceptance of the fact that uncertainty was a part and parcel of the system development. Although, more recently, the attention has been focused on issues related to managing the actual project, more and more empirical research is needed in every aspect of agile methodologies.

The SDLC can be described along a spectrum of agile to iterative to sequential. Agile methodologies, such as Extreme Programming (XP) and Scrum, focus on lightweight processes which allow for rapid changes (without necessarily following the pattern of SDLC approach) along the development cycle. Iterative methodologies, such as Rational Unified Process (RUP) and Dynamic Systems Development Method (DSDM), focus on limited project scope and expanding or improving products by multiple iterations. Sequential or big-design-up-front models, such as waterfall, focus on complete and correct planning to guide large projects, while minimizing risks to obtain successful and predictable results.

3.2.1 System Development Models

Regarding system development models, the more relevant are the waterfall, the iterative incremental and the agile development. These guide the design of the traditional methodologies in conventional system development, where the usual software product requirements are finalized before proceeding with the development.

Waterfall

The most commonly used software development model with this characteristic is the waterfall model[12]. However, in most of the cases, new functionalities get added and earlier requirements may change. Even though, the waterfall model is not structured to accommodate such continuous changes in requirements. One of its main characteristics is that one should move to a phase only when the preceding phase has been reviewed and verified. Therefore, the user does not have a clear sight on the functionality of the product until the product becomes available in its entirety.

Iterative Incremental

In the iterative incremental model[13], the development starts with a limited number of finalized and prioritized requirements, where the deliverable is a working increment of the product. A set of activities ranging from requirements to code development is called an iteration and, based on the functionality of the increment, the next lot of requirements is given to the subsequent iteration (Fig. 3.1). Following, the outcome of the subsequent iteration is an enhanced working increment of the product. This is repeated until the product accomplishes the required functionalities. The exact number and nature of the particular incremental builds, and what is iterated, will be specific to each individual development effort.

The basic idea behind this method is to develop a system through iterative cycles and in smaller incremental portions at a time, where both the development and use of the system contribute with

[12] http://en.wikipedia.org/wiki/Waterfall_model
[13] http://en.wikipedia.org/wiki/Iterative_and_incremental_development

lessons to improve the next iterations increments. Iterative and incremental developments are essential parts of the various agile software development methodologies.

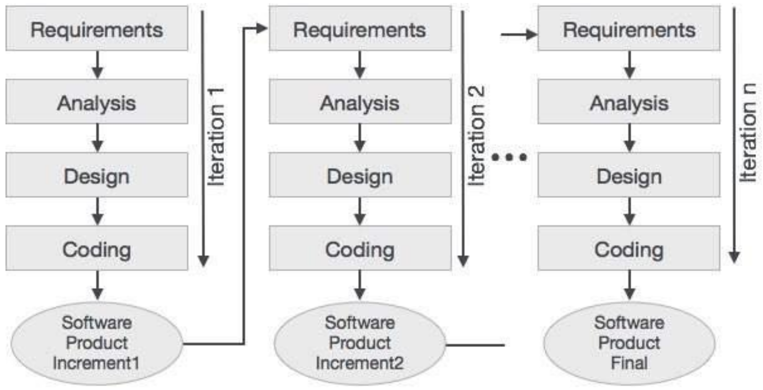


Fig. 3.1: Iterative Incremental Model

Agile Development

Agile development is based on iterative incremental development, in which requirements and solutions evolve through team collaboration, where a time-boxed iterative approach is recommended, so encouraging a rapid and flexible response to change. It is a theoretical framework and does not specify any particular practice that a development team should follow[14].

Early implementations of agile methods include, among others, the RUP (1994), Scrum (1995), XP (1996) and DSDM (1995). These are now collectively referred to as agile methodologies, after the Agile Manifesto that was published in 2001[15]. It was published by a team of software developers for highlighting the importance that needs to be given to the development team, while accommodating changing requirements and customer involvement.

3.2.2 System Development Methodologies

A system development methodology is a structured collection of best practices, guidelines and tools for development of a solution, usually in the form of a software product or a set of services. The most popular methodologies nowadays are OpenMethod and RUP, which are similar to each other, and XP and Scrum as the most agile and utilized examples. On a larger perspective, the Enterprise Unified Process (EUP) extends the system development lifecycle to cover the entire information systems domain.

Scrum

Within the agile methodologies, Scrum is a specific agile process framework that defines the practices required to be followed. It is the most popular agile framework, which concentrates particularly on how to manage tasks within a team-based development environment. Scrum also uses iterative and incremental development model, but with shorter duration of iterations. It is relatively simple to implement and focuses on quick and frequent deliveries[16].

---

[14] http://en.wikipedia.org/wiki/Agile_software_development
[15] http://www.agilemanifesto.org
[16] http://en.wikipedia.org/wiki/Scrum_(software_development)

Over the last 10 years, there was an ever-increasing volume of success stories, where companies have dramatically improved the success and performance of their information systems development teams and projects by recurring to agile practices. This has caused agile to be widely adopted across a variety of industries, including media and technology, large corporations and even government. The agile framework helps teams to benefit from faster time to market, in reducing the uncertainty and risk, and in increasing return on investment (ROI) by focusing on customer value. Among other different agile methodologies, Scrum has proved to be extremely successful worldwide over the last 20 years.

RUP – Rational Unified Process

The RUP is an iterative software development process framework, developed by Rational Software and owned by IBM. RUP is one of the most popular and complete process models used by developers in recent years. It is also the best-known and most extensively documented commercial variant of the Unified Process[17].

The elements that define the RUP are an underlying set of philosophies and principles for successful software development, a framework of reusable method content and process building blocks, and its underlying method and process definition language. To handle software development risks, RUP has established strategies to develop iteratively, manage requirements, use component architecture, model visually, verify quality continuously and manage changes.

RUP divides the software development lifecycle into two dimensions, phases and disciplines. The phases of RUP are inception, elaboration, construction and transition. As shown in the upper part of Fig. 3.2, phases divide the process over time and disciplines logically group activities by nature. Disciplines can last over all the phases and have different time consumptions. The nine disciplines make the vertical dimension, where each has its own activities, tasks, work products and guidance. Every phase is characterized by one or more iterations and each is concluded by a major milestone; when phase objectives are met the project moves to the next phase.

RUP's strengths include well-defined work products, easy combination with techniques from other methods and easy customization. RUP is also widely adopted, with good learning and consulting resources, and has what is needed for software development, although it seems a complex methodology with the enormous amount of information that can easily make it hard to use.

EUP – Enterprise Unified Process

The EUP is an extension to the RUP, so people familiar with RUP can see that the extensions include two new phases (lower part of Fig. 3.2), Production and Retirement, and several new disciplines: Operations and Support, and the enterprise disciplines: Enterprise Business Modeling, Portfolio Management, Enterprise Architecture, Strategic Reuse, People Management, Enterprise Administration and Software Process Improvement[18].

RUP defines a software development lifecycle and the EUP extends it to cover the entire information systems domain. Expected benefits are reducing information systems costs, improving business/IS alignment and improving business planning [Ambler, Nalbone, & Vizdos, 2005]. This extension to drive

[17] http://en.wikipedia.org/wiki/IBM_Rational_Unified_Process
[18] http://www.enterpriseunifiedprocess.com/

improvements across the entire information systems domain is justified as RUP is a powerful tool but does not go far enough, so, there is a need to extend it, including the cross-project and enterprise issues it largely ignores. The EUP encompasses that, enabling to deliver systems that meet all the needs of today's businesses.

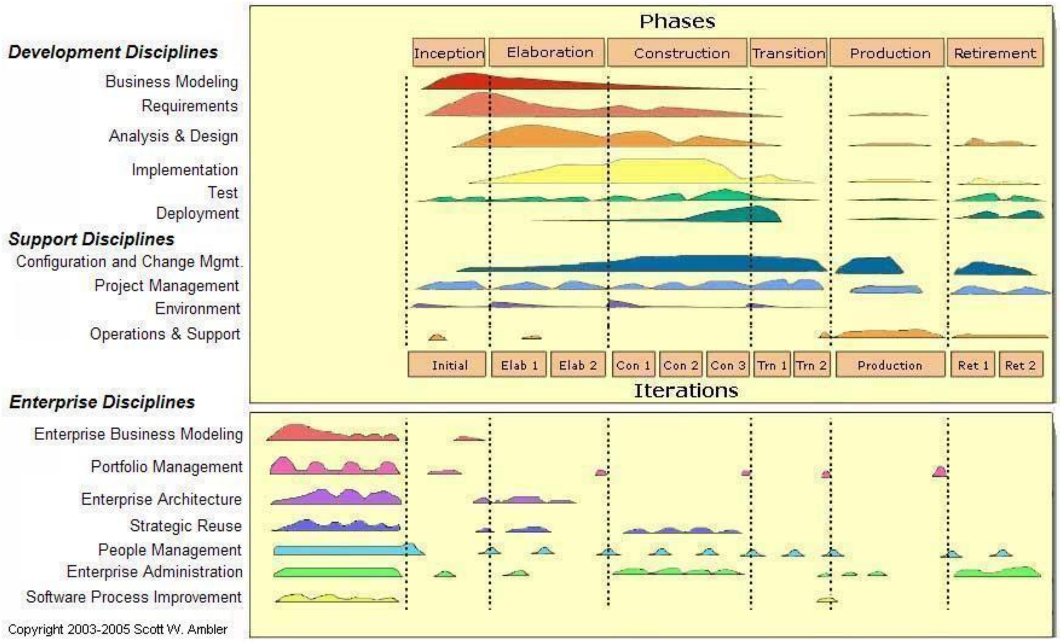


Fig. 3.2: EUP lifecycle

The EUP systematically identifies the business and technical problems that RUP fails to address. It introduces processes and disciplines for producing new software, implementing strategic reuse, "sunsetting" obsolete code and systems, managing software portfolios and more. Adopting the EUP requires the successfully mastering of RUP or another software development process, having enterprises issues to deal with, a need to operate and support the software after it is deployed or retire existing applications, and the willingness to invest in software process improvement.

3.2.3 System Development Methods

Organizations deal with the cost and time required to adopt agile strategies, either by starting out with agile and hoping to improve the chances of success by learning from other experiences, or already part way into agile transformation but struggling to figure out how it all fits together. Others are troubled with right-sizing their development process to meet the needs of a unique situation or project. These represent two common anti-patterns, whether starting with a process framework that is much too large or with one that is far too small. In order to avoid these extremes, organizations should instead take a middle-ground approach by starting with a method that is much closer to their actual needs[19].

The first process right-sizing anti-pattern is to start with a large process repository, the classic example being IBM's RUP. Although RUP is much maligned within the agile community the fact is that, if examined with an open mind, there are many very good ideas promoted by RUP. At the

[19] http://www.disciplinedagiledelivery.com/tag/rup/

other extreme is the right-sizing anti-pattern of starting with a small methodology, the classic example in this case being Scrum. With Scrum the idea is to add-in the techniques that Scrum does not address to right-size the process, but because Scrum proves to be only a very small part of the overall picture, to do this requires again significant process expertise, time and money.

Both of these anti-patterns represent extremes, starting with something large and cut it down to size, or starting with something small and build it up to meet the needs. In alternative, middle ground approaches benefit from starting with a tailored method with influences that fit closer to the actual needs, while considering a more pragmatic strategy. Addressing the full delivery lifecycle with different teams or different situations, and in order to avoid the bloatware of RUP or filling in the numerous blanks left by Scrum, it requires starting with a lightweight method and some tailoring advice. This sort of pragmatic thinking requires experienced developers, who do not just want to jump in and start coding, but rather to broaden their focus and work as a team in what is best for their organization.

Recently, within our research group, a V-Model based approach (Fig. 3.3) to derive logical architectural models from a process-level perspective, instead of the traditional product-level perspective has been proposed [Ferreira *et al.*, 2013]. This V-Model approach supports a method to encompass the initial definition of the project goals through organizational configurations, and the analysis and design of artifacts that result in a process-level perspective of the system's logical architecture. It then originates a validated process level structure and behavior architectural models that create a context for eliciting requirements, especially in difficult, ill-defined contexts.

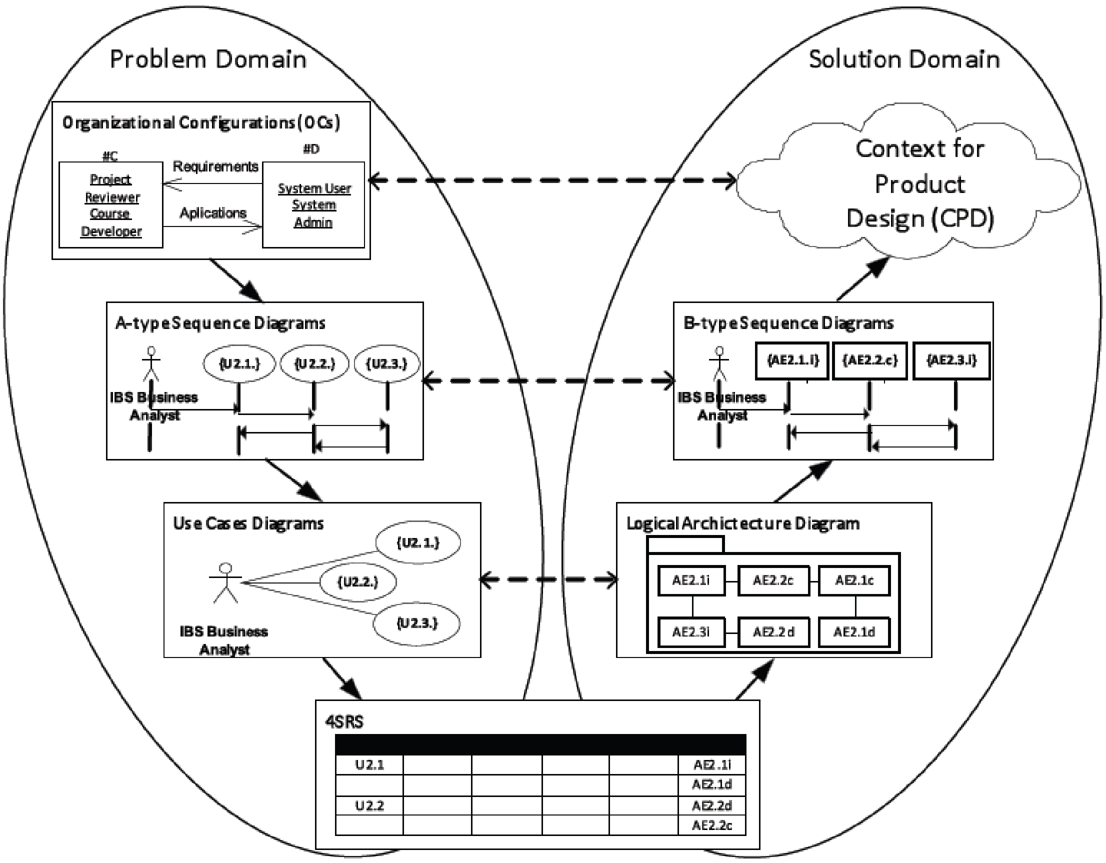


Fig. 3.3: V-Model Adaption for Business/IS Alignment [Ferreira *et al.*, 2013]

Instead of an architecture based on user requirements traditionally defined in a product-level perspective, this V-Model proposes the use of a process level perspective for the requirements definition and design of the logical model for the system architecture. It represents an information system by its logical architecture in order to achieve a structured view of its functionalities. This results from a process of transforming business-level and technological-level decisions, and requirements, into a representation (model) which is fundamental to analyze and validate a system, but is not enough for achieving a full transformation of the requirements into a model able to implement business stakeholders' decisions. To achieve such representativeness, there are added artifacts that represent requirements at different levels of abstraction, and promote an alignment between them and with the logical architecture.

The proposed V-Model approach is framed in the Analysis phase of the SDLC model, according to traditional development processes. It uses Organizational Configurations (OC), A-type sequence diagrams and use-case models, artifacts generated based on the rationale and in the information existing in previously defined artifacts, where A-type sequence diagrams are based on OC and use-case models are based on A-type sequence diagrams. The OC model is a high-level representation of the activities (interactions) that exist between the business-level entities of a given domain, with the intention of describing a feasible scenario that fulfills a business vision. A-type sequence diagrams, a UML stereotyped sequence diagram representation, describe the interactions gathered in early analysis phase of system development and are used afterwards as an elicitation technique for modeling the use-cases.

Next, the use-cases are modeled and textually described, and used as input for another in-house method of our research group, the four-step-rule-set (4SRS) method. The execution of the 4SRS method results in a logical architecture with a direct relation with the process-level use-cases assured by the method's execution. The process continues, with models placed on the left hand side of the path representation properly aligned with the models placed on the right side. The final resulting set of transformations along the V-Model path provides artifacts properly aligned with the organization's business needs (and formalized through OC).

3.3 OMG METAMODEL SPECIFICATIONS

The need for common standards and specifications work in the real world, and the benefits they bring to business, society and the environment led to the creation of the ISO organization (ISO is derived from the Greek *isos*, meaning equal). Its international standards ensure that products and services conforming to them are safe, reliable and of good quality[20]. For business, they are strategic tools that reduce costs by minimizing waste and errors, increasing productivity, helping companies to access new markets, level the playing field for developing countries, and facilitate free and fair global trade. Its standards are developed by the people that need them, through a consensus process, where experts from all over the world develop the standards that are required by their sector, so reflecting a wealth of international experience and knowledge.

On a more specific technology-oriented vision, the Object Management Group (OMG)[21], an international, open membership, not-for-profit technology standards consortium, deploys standards

[20] http://www.iso.org/

[21] http://www.omg.org/

driven by vendors, end-users, academic institutions and government agencies. OMG Task Forces develop enterprise integration standards for a wide range of technologies and industries, from their base specification to each respective implementation. It has been known for being at the core of the process of defining requirements and quickly developing specifications that are available commercially or open-source, acting proactively for them to become de facto standards.

Recent focus is on modeling (programs, systems and business processes) and model-based standards, including the Unified Modeling Language (UML) and Model Driven Architecture (MDA). Accordingly, this is supported with mechanisms to enforce interoperability, enable powerful visual design, execution, and maintenance of software and other processes.

As a solid business model, describing the business strategy, a business vocabulary and business rules, as well as business processes, play an essential role when developing an information system, our interest lies in the OMG Business Modeling Specifications. More specifically in the Organization Structure Metamodel (OSM), the Semantics of Business Vocabularies and Rules (SBVR), the Business Motivation Model (BMM) and the Business Process Modeling Notation (BPMN). This first four set of specifications standardize such business models. Although OSM is still not an official specification, there is already some relevant research with its first specification proposals and it has its placeholder in BMM to integrate with the other specifications.

Our attention directs also to two of other OMG Modeling Specifications, namely the Service oriented architecture Modeling Language (SoaML) and the Software Process Engineering Metamodel (SPEM). While SoaML is an interesting solution for a larger enterprise description of an information system, SPEM is oriented to the development process cycle. While SoaML complements the business-oriented aggregated structure of BMM and the three other specifications, SPEM can model and interact with the methodologies and tools that will be used in our proposals.

OMG also hosts organizations such as the user-driven information-sharing Cloud Standards Customer Council (CSCC) and the IT industry software quality standardization group, the Consortium for IT Software Quality (CISQ)[22]. As there is a growing sense of unease about the quality of software being managed by IT organizations for the benefit of their respective business operations, CISQ developed several software quality measures to analyze and manage the structural quality of IT application software, which are now OMG Approved Specifications, making them global standards for use in IT organizations.

Establishing a global standard for software quality is an important step for applications comparisons, which is essential in managing application development and maintenance towards better business outcomes. These standards are meant to complement and work in concert with process standards such as CMMI and ITIL, with a focus on the engineering attributes of software products.

3.3.1 Business Modeling Specifications

Within its organization around domain models, the OMG manages a number of standards for business modeling, including the Business Motivation Model (BMM), the Business Process Model and Notation (BPMN) and the Semantics of Business Vocabulary and Business Rules (SBVR)

[22] http://it-cisq.org/

specifications. Added to these, the Organization Structure Metamodel (OSM) is in a revised submission status, which means that it is not a complete specification yet.

BMM – Business Motivation Model

The BMM is a central piece on the OMG model and management specifications. It was initially developed by the Business Rules Group[23], with a first specification, 1.0, released in 2008. Later releases, 1.1 and 1.2, from 2010 and 2014 respectively, strengthened the integration with processes and rules, as well as the definition and refinement of some concepts. Recently, its latest version, 1.3, from May 2015[24], served merely for adding a note in the offering concept.

Together with the BMM, specifications such as BPMN, SBVR and OSM can be merged into a single business-oriented modeling architecture, and even implemented in integrated tool suites. Due to this centric-orientation, a solution can provide a straightforward way of relating business processes, business rules and organization units to each other, and to the desired results, courses of action and business policies that affect them.

The roles in the structure for those three essential concepts: Business Process, Business Rule, and Organization Unit, associate within the Business Motivation Model, but are regarded as references to elements to be defined and maintained outside an enterprise's BMM, within the previous referentials.

There are two major areas of the BMM, first the Ends and Means of business plans, and then the Assessments and Influencers. The first represent the 'being' and 'doing', where the Ends are things the enterprise wishes to achieve (Vision into Goals and Objectives) and the Means are things the enterprise will employ to achieve those Ends (Mission into Strategies, Tactics, Business Policies and Business Rules). The Influencers, who shape the elements of the business plans, represent its strengths and weaknesses, and the Assessments, which measure the impacts of the Influencers on Ends and Means, the opportunities and threats.

All elements of the BMM are developed from a business perspective, in order to develop a business model for the elements of the business plans before system design or technical development is initiated. That way, business plans can support the subsequent activities, connecting system solutions to their business intent. BMM brings benefits to developers of business plans, by defining a set of concepts or check-list of factors to be considered, a standard vocabulary and a flexible model to support development processes. Also, to business modelers, BMM helps in guiding and shaping more detailed models and functional tools, and to implementers of tools and repositories, it provides a notation with additional attributes, more-normalized entities and refined associations, as well as specifications of tools that support models with a broader scope.

BPMN – Business Process Model and Notation

The last version, 2.0.2, was recently released in December 2013[25], counting with minor tweaks from the previous version, 2.0, released in January 2011. The primary goal of BPMN is to provide a notation that is readily understandable by all business users, from the business analysts to the

[23] http://www.businessrulesgroup.org/bmm.shtml
[24] http://www.omg.org/spec/BMM/1.3/
[25] http://www.omg.org/spec/BPMN/2.0.2/

technical developers and finally to the business people. So, it creates a standardized bridge for the gap between the business process design and the process implementation. Another goal is to ensure that XML languages, designed for the execution of business processes, can be visualized with a business-oriented notation.

BPMN is designed for business process modeling and provides a graphical notation for specifying business processes, based on a flowcharting technique very similar to activity diagrams from UML. The primary goal of BPMN is to provide a standard notation readily understandable by all business stakeholders. It tries to unify the expression of basic business process concepts (public and private processes, or choreographies), as well as advanced process concepts (exception handling, or transaction compensation).

Nevertheless, BPMN is not without critics and weaknesses, usually related to ambiguity and confusion in sharing models, support for routine and knowledge work, and in converting BPMN models to different executable environments. Also, there are claims that it fails: to guarantee that standard-conforming business process models are interoperable; to constitute a basis for reliable communication of those models between different stakeholders; and to be implementable with the claimed 'generality' without restricting numerous 'variation points' by imposing semantically significant restrictions of concepts which are left underspecified [Börger, 2012].

OSM – Organization Structure Metamodel

The organizational structure is a fundamental aspect of the design of an enterprise. Traditional organizations relied primarily on a hierarchical structure while modern enterprises have a variety of relationships in order to operate more efficiently and responsively[26]. As referred previously, OSM it is not a specification yet, its status is of an Initial Submission.

OSM provides the concepts behind the organization of diverse and mutually independent corporate and government entities, enabling a metamodel to provide the basis for exchanging organizational models. Also, it provides connection points into several business components, ensures specifications to accommodate diversity in modern organizations, supports scenarios in which organizations merge, and consequently the merging of models developed in accord with the proposed metamodel. Moreover, consistency with leading conceptualizations of business rules and business process is important, because organization is typically defined in terms of rules and of roles and responsibilities, respectively.

The fundamental element of an organization structure is the Organization Unit, which are linked by Relationships. Other nuclear element is Position, which defines a Role in an Organization, where Organization Units may have a number of different Positions. Legal Entities are People and Corporations that participate in the enterprise, while Corporations are Organizational Units with related Organizational Units and Positions with People assigned, being People linked to Positions through Assignments.

[26] http://www.omg.org/cgi-bin/doc?bmi/09-08-02

SBVR – Semantics of Business Vocabulary and Business Rules

SBVR defines the vocabulary and rules for documenting the semantics of business vocabularies, business facts and business rules. Also in its aim is an XMI schema for the interchange of business vocabularies and business rules among organizations and between software tools. After the first version, 1.0 from January 2008, the following versions, 1.1 and 1.2, took 5 years to mature and be released. Recently, another significant revision led to the release of version 1.3[27].

This specification is intended to be the basis for formal and detailed natural language declarative description of a complex entity, such as a business, in order to formalize complex compliance rules, such as operational rules for an enterprise, security policy, standard compliance or regulatory compliance rules, which can be interpreted and used by computer systems[28].

SBVR allows the production of business vocabularies plus rules, which constitute a shared domain model with the same expressive power of standard ontological languages. It enables making business rules accessible to software tools, including tools that support the business experts in creating, finding, validating and managing business rules, and the information technology experts in converting business rules into implementation rules for automated systems. Although it supports multilingual development, as it is based on the separation between symbols and their meaning, it proposes Structured English as one of the possibly many notations that can map to the SBVR metamodel.

In practitioners view, SBVR still has some limitations, as many of the business rules specified by business people should be automatically executed on the underlying information system, in order to bridge the gap between UML (namely with OCL expressions) and SBVR. By providing an automatic transformation from OCL to SBVR specifications, it allows for designers to be able to interact with the business people, in so refining and validating the information modeled in the conceptual schema before the generation of the final information system implementation. This is possible by overcoming some of the existing differences between the two specifications, proposing some extensions to SBVR that facilitate the translation from UML/OCL to SBVR specifications [Cabot, Pau, & Raventós, 2010].

3.3.2 Modeling and Metadata Specifications

On another domain the OMG manages a number of standards for modeling and metadata, including the Service oriented architecture Modeling Language (SoaML) and the Software and Systems Process Engineering Metamodel (SPEM). Also related to these, the Software Quality Characteristics specification is a joint collaboration with the Consortium for IT Software Quality (CISQ).

SoaML – Service oriented architecture Modeling Language

SoaML provides a metamodel and a UML profile for the specification and design of services within a service-oriented architecture, its latest version, 1.0.1, was released in May 2012[29]. It embraces and

[27] http://www.omg.org/spec/SBVR/1.3/
[28] http://en.wikipedia.org/wiki/Sbvr
[29] http://www.omg.org/spec/SoaML/1.0.1

exploits technology as a means to an end but is not limited to technology architecture, enabling business and systems oriented services architectures to mutually and collaboratively support the enterprise mission.

The existing models and metamodels for describing system architectures have not been sufficient to describe SOA in a precise and standardized way, with UML itself being too general for the purpose of describing it, while needing clarification and standardization of even basic terms like provider, consumer, etc. SoaML has been created to support the modeling capabilities for identifying services and its dependencies to services requirements, also for specifying services and defining its consumers and providers, and even the policies for using and providing services, as well as its classification schemes.

It supports integration with BMM and further extensions related to integration with BPMN, SBVR, OSM and others. Particularly, it allows linking the business vision, goals, strategies, tactics as well as business rules according to BMM, then bridging the resulting business specifications toward components of a Service Oriented Architecture (SOA). It connects BMM with SOA, aligning the system's processes according to its goals and directives, while most of the emerging SOA approaches try to determine their services on the basis of business processes, being totally disconnected from business goals and rules [Berkem, 2008].

Moreover, in order to support a swiftly and coherently reaction to changes, an agile SOA architecture should integrate the core elements of the BMM, providing organizations with the capability to capture how services realize business motivations. This allows increasing their competitiveness by synchronizing their information systems with the evolutions of their business goals and directives, so structuring their architecture according to business motivations. Also, this supports the identification of traceability relationships that allow connecting such high level business goals and directives toward elements of the software level specifications.

SPEM – Software and Systems Process Engineering Metamodel

The SPEM version 2.0, released in April 2008 and standardized by the OMG, is a process engineering metamodel that provides to process engineers a conceptual framework for "modeling, documenting, presenting, managing, interchanging, and enacting development methods and processes"[30]. It defines the semantics of this metamodel as well as its direct application for all method and process modeling activities. The goal is to accommodate a large range of processes, and not to exclude them by having too many features or constraints.

SPEM is used to define software and systems development processes and their components, limiting its scope to the minimal elements necessary, without adding specific features for particular development domains or disciplines. The goal is to try to accommodate a large range of development methods and processes of different styles, cultural backgrounds, levels of formalism, lifecycle models and communities, although its focus is on development projects. While not intending to be a completely generic modeling language, it contrarily provides only the minimal concepts needed to describe a development process.

[30] http://www.omg.org/spec/SPEM/2.0/

For many development approaches and methods, human consumable documentation providing understandable guidance for best development practices is more important than precise models. They are given a higher value than strict obedience to a formally defined process, as they cannot be formalized with models, but only captured in natural language documentation.

With SPEM, users can define Method Content, primarily expressed using work product, role and task definitions, and guidance in a general direction, building up a knowledge base of development methods. This supports development practitioners in setting-up a knowledge base of intellectual capital for software and systems development, which allows them to manage and deploy their content using a standardized format. Then, development teams are able to define how to apply the development methods and best practices throughout a project lifecycle, selecting and tailoring the development process as they require.

In a recent study [Dodero, Palomo-duarte, Ruiz, & Gawn, 2012], SPEM was considered the most widespread and popular language for representing development processes, with a high degree of acceptance of its metamodel, and its uses and applications. Nevertheless, over half of the papers collected only used SPEM as an annotation to represent certain activities in the context of the research performed in each case, while many others described extensions for improving certain deficiencies in the SPEM metamodel. Despite these weaknesses, SPEM is considered as a suitable language for representing development processes.

Summarizing, SPEM is a process engineering metamodel as well as conceptual framework that provides the necessary concepts for modeling, documenting, presenting, managing, interchanging and enacting development methods and processes. It is targeted at process engineers, project leads, project and program managers who are responsible for maintaining and implementing processes for their development organizations or individual projects.

CISQ – Software Quality Characteristics

ISO 25010 defines the Quality Characteristics (QC) for Software Systems, with QC being composed by several quality sub-characteristics, each consisting of a collection of quality attributes that can be quantified as Quality Measure Elements (QME). These QME can either be counts of structural components or violations of rules of good architectural or coding practice. During the course of several expert forums, four main QC were selected as the most important targets for automation, namely: Reliability, Security, Maintainability, and Performance Efficiency. These specific targets cover half of the eight QC described in the complete ISO 25010.

Each QC is decomposed into a set of issues, each issue is decomposed into a set of rules and each rule is translated into a violation. All these measures can be used in evaluating and managing information systems business applications, as the work leading to these specifications is to provide international standard definitions against which organizations, service providers and software vendors can implement automated measurement of the structural quality of their software.

Accordingly, the Consortium for IT Software Quality (CISQ) recently released the Specifications for Automated Quality Characteristic Measures[31], a specification for automating the measurement of those four Software QC. These are consistent with the definitions given for each QC in ISO/IEC

[31] http://it-cisq.org/standards/

25010, where measures for each QC aggregate counts on violations of rules for good architecture and coding practice. Also consistent with the evolving ISO/IEC 25000 series of standards, the System and software Quality Requirements and Evaluation (SQuaRE), Software QC are increasingly being incorporated into development and outsourcing contracts as the equivalent of Service Level Agreements (SLA). These provide common measurement definitions and calculations for the four referred Software QC, as prioritized by CISQ members.

3.4 SUPPORT TOOL ENVIRONMENTS

Within a call to align metamodeling with formal language engineering techniques, MDA aims to exploit the usefulness of models as tools for abstraction, by focusing on architecture, artifacts and models. This succeeding, tooling is essential to maximize the benefits of having models, be it in modeling different dimensions, by mapping its diverse perspectives, or in modeling the transformations between them [Kent, 2002]. An important aspect of realizing such tools is to ensure that they remain flexible and configurable, as their aim is within different domains, often in different organizations and sometimes in different projects. This way, a model driven engineering approach can better support specifying the modeling languages, models, translations between models and languages, and the process used to coordinate the construction and evolution of the models.

Also, as there is some commonality between these processes, it follows that it may be possible to define families of processes, with the desirable result of making it easier to develop/configure tools to different processes. Metamodeling, which has been used to define the abstract syntax of a modeling language, can show work that allows concrete syntax and semantics to be defined using a metamodeling approach. Moreover, with regard to defining processes, there is also the contribution of the UML profile for defining software development processes, the Software Process Engineering Management (SPEM) UML Profile.

Following this necessity for tools and in respect to our work, we will be regarding this dual side of tools, either supporting the dimensions modeling and its transformations, and also on the perspective of processes configuration. As the focus is on the OMG specifications and these are all UML-based, we will concentrate on the most popular UML modeling tools available. Also, as the span of our work is limited in time and resources, conditioning the development of full-fledged tools, we will be looking for recent, lightweight, trends for prototype development environments.

3.4.1 UML Modeling tools

In order to specify, design and handle models, a tool that supports a range of operations on them (definition, modification, transformation, cross-modeling, etc.) is needed. A brief look at the list of the available UML tools, focusing in the open-source ones, directs our attention to the Eclipse UML2 tools, from the Eclipse foundation, as a reference in the field. On the other hand, outside the open-source market, the Enterprise Architect (EA) from SPARX Systems and the Rational Rhapsody (RR) family from IBM, stand out as very complete and interesting solutions in their own way. Regarding the previously referred development methodologies, Scrum, EUP and RUP, this last one through the IBM Rational Method Composer (RMC), all are included in the Eclipse Process Framework (EPF) built on top of Eclipse [Terävä, 2007].

On a quick side-by-side on these solutions, as general characteristics, the RR are based on the Eclipse Modeling Framework (EMF) and the UML2 layer, which means they are in the Eclipse ecosystem and can run its tools on most platforms, allowing to use any EMF and Eclipse plugins to enhance them. The downside with Eclipse is its guided user interface (GUI), which is based on standard widget tool (SWT), feels heavy and slow. On the other hand EA is Windows-native, but also supported on Linux, with its GUI feeling like any Windows program, quicker and smoother. Also, while RR feels more cumbersome and obsessed with details, EA presents a more careless and open-minded attitude.

EA's plugin architecture is proprietary, so plugins are specifically developed for it, while EMF/UML2 provides a higher degree of UML conformance than EA's database-based data model. Both RR and EA support model-to-model transformations, but most specific transformations are not supported and have to be constructed by the users themselves. Both tools can be extended, RR through the Eclipse plugin architecture and Java, EA through a proprietary model-based mechanism and .NET. As it is difficult to find any tool that suits ones specific needs out of the box, usually adaptations will have to be made, as also changes to ones' internal processes. EA is the solution presenting more modeling power, functionality and ease of use, while RR ensures a higher degree of UML conformance and for using Eclipse-based tools[32].

Eclipse Framework

The Eclipse Framework supports the Eclipse Modeling Tools package which contains a framework and tools to leverage models, with a Java/XML framework for generating tools and other applications based on simple class models. It is intended to provide the benefits of formal modeling, helping in turning models into efficient, correct, and customizable Java code, ready for easy tool deployment[33]. The Eclipse Framework has available different download packages for different development purposes and plugins that can be added to expand its functionality. One of its more interesting packages is the Eclipse Modeling Tools, a medium-size popular solution containing a framework and tools to leverage models, including a complete SDK, developer tools and source code.

Eclipse is an open source community, whose projects are focused on building an extensible development platform, runtimes and application frameworks for building, deploying and managing software across the entire software lifecycle, with strong emphasis on Java IDE. Its community is also supported by a large ecosystem of major IT solution providers, innovative start-ups, universities and research institutions and individuals that extend support and complement the Eclipse Platform.

The Eclipse modeling technologies are not just about software development, they harbor projects in several industries and academic research, as where there is anything significant to work focused on manipulating data, modeling will play at least a small part, since models play a ubiquitous role. The EMF project is a subproject of the top-level Eclipse Modeling project, being used by a large and growing number of other projects at Eclipse. All these demand a top-level Modeling project to oversee them, so the Modeling project is like an onion with many layers and EMF at its core.

[32] http://stackoverflow.com/questions/9775717/rational-ea-or-something-else
[33] http://help.eclipse.org/juno/index.jsp

Models can be created using annotated Java, XML documents or modeling tools like Rational Rose, then imported into EMF and the code generator turns a model into a set of Java implementation classes. These classes are extensible and regenerable, that is the Java code can be used to update the model, which allows generating a completely working editor from a picture of the object model, and running the generated editor to add and remove elements. This illustrates the power of MDA, just by providing a model for the data that an application will manipulate. Since many Java developers are familiar with simple class modeling, this serves as a 'gentle introduction' to UML class modeling, allowing the benefits of model based code generation to a larger community of developers.

Another interesting solution under the Eclipse umbrella, one of the Eclipse Technology Projects, is the EPF. The EPF aims at producing a customizable software process engineering framework, with process content and tools, supporting a broad variety of project types and development styles. EPF uses SPEM and although the title implies Software Processes, any process can be represented using SPEM.

Sparx Enterprise Architect

Sparx Systems Enterprise Architect is a visual modeling and design tool based on OMG UML. This platform supports the design and construction of software systems, and the modeling of business processes and industry based domains. It is used by businesses and organizations to not only model the architecture of their systems, but to process the implementation of these models across the full application development lifecycle[34]. By using UML for systems modeling, it provides a basis for modeling all aspects of organizational architecture, along with the ability to provide a foundation for designing and implementing new systems, or changing existing systems.

Along with system modeling, EA covers the core aspects of the application development lifecycle, from requirements management through to design, construction, testing and maintenance phases. It counts also with support for traceability, project management and change control of these processes, as well as facilities for model-driven development of application code using an internal integrated-development platform. The user-base ranges from programmers and business analysts through to enterprise architects, in organizations ranging from small developer companies, multi-national corporations and government organizations through to international industry standards bodies.

Sparx Systems initially released EA in 2000. Originally designed as a UML modeling tool for modeling UML 1.1, the product has evolved to include other OMG UML specifications until the latest UML 2.5. Enterprise Architect supports a range of open industry standards for designing and modeling software and business systems, among which: BPMN 2.0, SoaML and SPEM. EA also supports frameworks such as Zachman and TOGAF. Relating to underlying UML modeling, there are several key aspects that this modeling tool supports, including: Profiles, Patterns, MOF, OCL, MDA Transforms, Corba IDL and UML Validation (which can be run against the model); among the available Microsoft Visual Studio .Net and Eclipse plugin interfaces.

[34] https://en.wikipedia.org/wiki/Enterprise_Architect_(software)

IBM Rational Rhapsody family

The Rational Rhapsody family gathers the IBM Software products for modeling and simulation, under their Continuous engineering unit, where systems and software engineering provide the foundation for product and systems development, offering the agility to develop products and product lines in an open, connected, continuous engineering environment. They provide tailored solutions within a collaborative design, development and test environment, for systems engineers and software developers, allowing for rapid prototyping and execution to address errors earlier when they are least costly to fix. They count also with automatic consistency checking to enhance agility and improve reuse, with collaboration to reduce both recurring and non-recurring costs, and sharing, collaborating and reviewing engineering artifacts, created with Rational Rhapsody or other design tools for the entire development lifecycle[35].

This family of tools allows system engineers to analyze, execute and validate requirements to specify robust architectures, while enabling real-time and embedded software development, being agile and automatically generating fully executable code while complying with safety standards. It also helps organizations to create connected products and systems aimed at the cloud, where these enterprise solutions help to increase product value through engineering practices aligned to the accelerating pace of business change. Although it is a commercial, expensive, product, it also presents free solutions for research and classroom use.

This family edition highlights the products:

- Architect for Systems Engineers, a system engineering environment for analyzing and elaborating requirements, making architecture tradeoffs and documenting designs;
- Architect for Software, an integrated software engineering environment to graphically architect applications using UML, reuse existing code with reverse engineering, generate code frames, automatically synchronize design and code, generate design documentation and graphically specify test cases;
- Design Manager, enabling cross discipline teams to collaborate, share, review and manage design information across the development lifecycle;
- Designer for Systems Engineers, including all the capabilities of Architect for Systems Engineers plus the ability to prototype, simulate and execute designs for early validation of requirements, architecture and behavior;
- Developer, an embedded and real-time agile software engineering environment with application generation, rapid prototyping and simulation for design level debugging, automated build generation for continuous integration and support for safety critical software lifecycle, in addition to the capabilities of Architect for Software.

3.4.2 Prototype development

These former solutions are very powerful, allowing to follow the complete SDLC, while integrating all steps to cover the entire organization. Nevertheless, due to their size and complexity, which comes associated with a heavy learning curve, they demand a complete team and full-time dedication in order to develop a working solution. This due, within the viable context and limited timespan of our work, and in order to deliver the prototype tools we deemed important, the decision was to follow a

[35] http://www-03.ibm.com/software/products/en/ratirhapfami

recent trend in technology, through the development of small, agile application in the form of apps. This solution enables a fast delivery of products, with the possibility of continuous innovation, due to their easy maintenance and persistent feature upgrade.

As every business faces the reality to innovate or perish, regardless of industry, the need to deliver modern, multi-channel applications that engage customers and empower employees has never been more urgent. Nevertheless, growing project backlogs and unhappy business sponsors are indications that traditional development approaches do not answer these challenges. Enterprise application Platform-as-a-Service (aPaaS) offers a way forward, promising to accelerate application delivery cadence and capacity, and with several aPaaS offers in the market (Fig. 3.4), it is difficult to understand and navigate the aPaaS landscape, and ultimately, select the platform best suited to ones organization's priorities[36].

Fig. 3.4: Magic Quadrant for Enterprise Application Platform as a Service

As application platform technology innovation moves to the cloud, the CIO, IT planners and architects, driven by demands of cloud, mobile and internet of things (IoT), seek that innovation in technology and business. Application infrastructure functionality, enriched with cloud characteristics

[36] http://ww2.mendix.com/rs/mendix/images/gartner-magic-quadrant-2015-apaas-report.pdf

and offered as a service, is named platform as a service (PaaS), more precisely as cloud application infrastructure services.

Application platform as a service (aPaaS) is a form of PaaS that provides a platform to support application development, deployment and execution in the cloud. It is a suite of cloud services designed to meet the prevailing application design requirements of the time, and nowadays it includes mobile, cloud, IoT and big data analytics innovations. We point out three of the most prominent vendor offerings designed to support and advance these initiatives: IBM Bluemix, Mendix Platform and OutSystems Studio.

IBM Bluemix

IBM's aPaaS offering is part of its Bluemix suite of PaaS capabilities (xPaaS), developed and supported by IBM, its partners for the open community. These include database, integration, mobile, DevOps, security, monitoring and other services. IBM Bluemix is deployed on the network of SoftLayer data centers worldwide, being based in part on the open-source OpenStack and Cloud Foundry v.2 software[37].

The Bluemix Liberty for Java offering implements IBM WebSphere Application Server Liberty Core as a Cloud Foundry buildpack. Managed by the Cloud Foundry PaaS framework, it is a shared-OS, high-control, cloud-based application platform service, also providing an SDK for Node.JS aPaaS environment, and other capabilities delivered as Cloud Foundry services. Other language environments and frameworks available on Bluemix include the community-provided PHP, Go, Ruby and Python, where users are also able to add their own language environments utilizing the open-source buildpack support in IBM Bluemix.

Mendix Platform

Mendix is a small, but well-established, pure-play aPaaS provider. The Mendix Platform is a shared-OS, multitenant aPaaS comprising a high-productivity development environment (Mendix Business Modeler), a collaboration and administration portal (Mendix Developer Portal), and a runtime server (Mendix Business Server). Moreover, the Mendix App Store provides a venue for vendors and users to share applications, widgets and services[38].

Mendix Business Modeler supports the "no-code" development of multichannel applications using visual, process-oriented, model-driven development and service composition. Models can be extended using code written in Java, Scala and JavaScript. The development environment is available as a cloud service, or it may be downloaded to a developer's computer. In either case, application metadata is stored in a cloud-resident repository, and applications can be deployed to any Mendix Business Server, which is also available as a Cloud Foundry buildpack.

OutSystems Studio

OutSystems is a high-productivity enterprise rapid application delivery (RAD) PaaS that focuses on accelerating the time to solution of enterprise apps. OutSystems' platform uses an indirectly executed metadata-driven model, which generates .Net or Java code that ultimately drives the

[37] http://www.ibm.com/cloud-computing/bluemix/
[38] http://www.mendix.com/

execution of the application. Applications are developed using native desktop tools and deployed to the on-premises, private cloud or public cloud, and are usable with Web and mobile devices[39].

The platform makes extensive use of metadata models to configure the application layers, enabling high-productivity development and faster time to solution. Developers can incorporate their own custom Java or C# code or libraries, and compose them as part of the model. The built applications are portable, can be exported to an Eclipse or Visual Studio project and can be deployed and maintained outside the OutSystems aPaaS.

Also, it allows developers to take a hybrid approach to application and hosting. Through the OutSystems Forge, anyone can provide prebuilt components and sample applications that offer different platform capabilities. It counts already with an interesting base number of sample applications and industry frameworks that demonstrate how different components can quickly be brought together to build an application.

3.5 CONCLUSION

The grounded set of terms for modeling in information systems development, due to its crucial importance and widespread use, the benefits for the use of standards and associated metamodels, namely the popular business modeling, service and process related OMG specifications, and the development of a set of lightweight prototype tools to aid and support the methods to be made available, set the basis on firm ground for the main contributions of our work.

Among the diverse system development models and methodologies, our choice follows on an agile method with strong influences from RUP, although avoiding the heavy procedures it imposes. Our selection follows on a middle ground approach, for a method developed within our research group, right-sized to our projects, the V-Model and its associated 4SRS. Besides the natural interest in working with methods developed within our own research group, these two methods were set as the basis for our work also because they constitute somewhat lightweight methods, supported on tailoring capabilities that address the full delivery lifecycle, which allow avoiding the bloatware of RUP and the filling in of the numerous blanks left by Scrum.

In what relates the OMG specifications, BMM stands as a central reference for establishing the connection between functional and nonfunctional sides, while SoaML can be an interesting bridge for the system architecture, along with CISQ software quality issues. On the other side, SPEM is directed at the definition of generic methods and tailorable processes within our proposals. These specifications were chosen directly as an intrinsic part of the methodologies that we use, in part due to the support of the OMG specifications on the development of meta-methods and tools, and also because standardized methods and modeling are an essential part of our work and vision.

Despite the existence of interesting and powerful solutions for the development of model-based support tools, as the Eclipse Framework, the IBM Rational family and the Sparx Enterprise Architect, the restrictions on the timespan and resources associated to this work led us to consider other solutions. As a lightweight option for the quick and easy development of prototypes, in the form of support tools for our proposed methods and processes, the OutSystems Studio stood out among

[39] http://www.outsystems.com/

other aPaaS development environments. The platform's model-based composition of applications and extensive use of metadata models, as well as its Portuguese origin and the main author's relationship with some of its core developers consolidated it as a firm choice among the available platforms.

Following these informed choices, also based on our personal and associated research team interests, our work continues based on the sustained evolution of the V-Model and associated 4SRS methods, counting on the OMG business modeling specifications and other previous referred reference models as the BSC and BMC, and with the support of SPEM and the OutSystems Studio, in order to develop and support our proposals.

—— This page intentionally left blank ——

CHAPTER 4

ELICITATION OF BUSINESS/IS REQUIREMENTS: THE PGR APPROACH

Summary: Model-based transformations play a central role in the development and maintenance of information systems, namely in software-based solutions based in elicited business requirements. After an initial comparison of our research groups' 4SRS with other proposals in the academia inspired in the Twin Peaks transformation model, we present our PGR proposal, relating functional and nonfunctional requirements, which is supported in a metamodel for its representation and a tailorable method for its manipulation. Associated to it we propose the development of a tool support, within the OutSystems Studio environment, with its entities structure, screen flows and form captions.

CHAPTER TABLE OF CONTENTS

PUBLICATIONS:

- MATRA: A Framework for Assessing Model-based Approaches on the Transformation between Requirements and Architecture (**EMMSAD'16, Springer**)
- Using Process-level Use Case Diagrams to Infer the Business Motivation Model with a RUP-based Approach (**ISD'13, Springer**)

—— This page intentionally left blank ——

CHAPTER 4

ELICITATION OF BUSINESS/IS REQUIREMENTS: THE PGR APPROACH

"Go on, then, you make the coffee while I try to bring some order to this chaos, and then the unexpected happened, for, as if giving no particular importance to the words emerging from her mouth or as if she did not entirely understand them, she murmured, 'Chaos is merely order waiting to be deciphered'"

– José Saramago, **The Double** (2002)

4.1 INTRODUCTION

Under the thematic of aligning information systems with business, considering the initial model-based development and continued maintenance of an information system, we initially prioritized a plethora of topics into three main problems. The first of them, which is the subject of this chapter, is what type of requirements should be used and how should they be elicited in order to set the basis for such an alignment solution. Accordingly, as previously referred in Chapter 2, handling requirements is all about managing the transition between problem-solution domains. This involves constantly adapting and incorporating business changes, as well as technologic breakthroughs, in the resulting information system, where, in our practice, software implements most of the system components.

Software Engineering (SE) is inherently a modeling activity in which abstract models of information systems are derived from Requirements Engineering (RE) and then systematically developed from problem to solution space [France & Rumpe, 2007]. The proficient use of models has been helping traditional engineering in achieving success and boosting product quality, nevertheless, in RE and SE that is not so perceptible. This is due to technical issues but also to a number of complex social and economical ones [Selic, 2012]. Even though the use of standards and models in RE and SE is noticeable, as researchers try to communicate clearly and contribute on common grounds while practitioners struggle to adopt existing proposals [Whittle, Hutchinson, & Rouncefield, 2014].

Model-driven development research aims to play a role in establishing and spreading the use of model-based approaches in RE and SE. In this context, the early step of transforming requirements into an analysis model is a crucial and challenging one. The Twin Peaks model [Nuseibeh, 2001], with its spiral life-cycle archetypal, stands as a reference in linking the RE and SE fields, setting the track for the transformation, traceability and alignment between the elicited business requirements and the candidate system architectures. Also, issues as quality characteristics, architecture evaluation and development of supporting tools have become essential elements for any proposed approach in this domain [Grünbacher, Egyed, & Medvidovic, 2004].

Inspired by the transformation process between these two worlds, numerous model-based proposals have been presented and evolved along the years, raising several issues in this subject [Galster, Eberlein, & Moussavi, 2009]. Among them, a well-grounded choice of an elicitation technique [Loniewski, Insfran, & Abrahão, 2010], accompanied by a clear description of the transformation, alignment and traceability mechanisms between requirements and its associated analysis model elements are seen as vital.

Moreover, questions about representational models for both requirements and architectures, evolution of the relations between business-oriented and system-oriented requirements, diversity in detailed process specifications and design, case study evaluations and support tool development, have been explored. The continued evolution and evaluation of all these issues, namely through empirical studies, is evermore necessary in order to advance research and influence practitioners adoption [Yue, Briand, & Labiche, 2011].

Grounded on the foundations of these previous works in the comparison and systematic review of existing transformation approaches, while supported on a set of recent, research relevant, model-based proposals, we present a framework covering their involved key issues. This framework allows classifying and categorizing the individual details of each approach, further facilitating the assessment and comparison of methods, as well as the respective artifacts involved in the different approaches. It also details and updates previous proposals results, again according to the latest research evolutions in this topic, with a focus on structural transformation approaches and in line with the influence of the Twin Peaks model.

The proposal of a method, within our own research group, for the derivation of a logical architecture from process requirements, previously referred in Chapter 3, the 4SRS [Ferreira, Santos, Machado, Fernandes, & Gasevic, 2014], follows in line with these issues. In order to assess our groups' proposal, we frame and compare it relatively to other similar and relevant model-based approaches in this domain, performing a review on their transformation issues from requirements to architectures.

This involves creating a framework covering each of the previously identified research issues and then classifying each of the considered approaches accordingly. Besides the 4SRS method, we selected the CBSP technique [Grünbacher *et al.*, 2004], the ATRIUM methodology [Montero & Navarro, 2009] and the STREAM process [Castro *et al.*, 2012] to take part in this study, a choice based on their recent associated research relevance and alignment with the Twin Peaks model.

After a first contextualization of the current problems and trends faced by these model-based transformations, we go over the referred approaches presenting them and reviewing all the issues involved in this topic. Next, we present the framework with the coverage for each approach on the relevant issues analyzed, while discussing the gathered results. This posed, afterwards we present our own proposal for an initial approach on the elicitation of requirements adequate to the business/IS alignment, covering the functional and nonfunctional sides of an information system. Next, the development of a tool support prototype for this approach associated representation metamodel and tailorable method is described, counting on its structure, process flows and available forms, and finally some conclusions are stated.

4.2 SYNOPSIS OF FUNDAMENTAL CONCEPTS

The presentation of the Twin Peaks model [Nuseibeh, 2001] allowed for a better understanding and articulation of the conceptual differences between requirements and design, while inspiring much research in both domains. Nevertheless, although its spiral life-cycle model stands as a reference in the requirements engineering and information systems architecture fields, the process of moving between the problem and solution worlds is not as well recognized.

To begin with, the relation between architecturally significant requirements and architectural design decisions is not always controversy free. In fact, there seems to be no fundamental distinction between them, as they can be perceived as being observed from different perspectives [de Boer & van Vliet, 2009]. A certain amount of creativity is always involved and there are different levels of perspective from diverse stakeholders, leading to ambiguity and whether to call something a requirement or an architectural design decision. As complementary and aligned approaches, one cannot do without the other.

Regarding the transformation process between the two worlds, traditional solutions as the CBSP approach [Grünbacher *et al.*, 2004] try to solve this issue in diverse ways. As requirements and architectures use different terms and concepts to capture the model elements relevant to each other, one solution is to relate and reconcile those using intermediate models. The process of reconciling is always a difficult task, much based on intuition and experience, where some automation through tool support is desirable, but not a full one, as human intervention is instrumentally decisive.

Also, gradually, quality issues have been the target of increased attention, with system architecture as a major determinant of system quality [Croll, 2008]. While functional properties determine what the software is able to do, the nonfunctional (quality) properties determine how well the software performs, where explicit architectural decisions can facilitate optimization among quality attributes. Standards like ISO/IEC 25010 help define quality attributes from both an internal and an in use perspective, addressing architectural design and system realization, respectively.

Moreover, architectural evaluation is becoming a familiar practice for developing quality software, as it reduces development efforts and costs. By verifying the addressability of quality requirements and identifying potential risks, it provides assurance to developers that their chosen architecture will meet both functional and nonfunctional quality requirements [Shanmugapriya & M. Suresh, 2012].

Standing as a reference in the field, the ATAM technique [Kazman *et al.*, 1998] supports the evaluation of architectures and architectural decision alternatives in light of quality attribute requirements. It takes proposed architectural approaches, analyzes them, and identifies sensitivity and tradeoff points, describing stakeholders' interaction with the system.

Alongside quality, the realities and necessities of modern software development acknowledges the need to develop architectures that are stable, yet adaptable, in the presence of changing requirements. The question of software evolvability, which describes a software system’s ability to easily accommodate future changes, makes evolvability a strong quality requirement in an ever-changing world. As business and technology progress, and software becomes more complex, development teams face the challenge on how to evolve the systems in their operationally changing contexts [Breivold, Crnkovic, & Larsson, 2012].

4.2.1 Model-based Transformations

Within the domain of the transformation from business requirements to system architectures, a number of diverse approaches have been proposed, which instigated some comparisons and systematic reviews. Among the approaches, the CBSP technique [Grünbacher *et al.*, 2004] stands as one early well-cited reference, directly following the Twin Peaks model.

Following those studies and in line with this influence, we identified three other recent and research relevant, model-based approaches. The early ATRIUM methodology [Montero & Navarro, 2009] and the more recent STREAM process [Castro *et al.*, 2012] present heavy-modeled solutions counting on diverse views, which have sparked interesting discussions in this research domain.

The latest proposal to include in this study is our own research groups' 4SRS method [Ferreira *et al.*, 2014], part of a V-Model method to derive logical architectural models, especially in complex or ill-defined contexts. Next, these four approaches are presented in more detail.

The CBSP technique

Directly following the Twin Peaks model, the CBSP technique [Grünbacher *et al.*, 2004] stands as an early reference for the transformation from requirements to architecture. Its taxonomies for both requirements and architecture representational models count with an additional intermediate model to iteratively evolve them. Also, its clearly defined transformation process supports the iterative, concurrent development of requirements, architectures and intermediate CBSP models.

With the associated case study and tool definition presented in the paper, this proposal represented a fairly complete solution at that time for a scalable and human intensive problem. Its work on refining requirements complements its process with a structured transformation technique and tool support, emphasizing a multi-perspective of requirements engineering and also on conflict detection and resolution. Nevertheless, it lacks a more formal treatment of requirements.

Its support for traceability eases capturing and tracing links, by narrowing the gap between informal or semi-formal requirements and architecture models, as the intermediate CBSP model also helps to relate architectural issues and requirements. With regards to recent trends, it lacks a deeper integration with standards reference models and evaluation of the resulting architecture. Although presenting itself as a simple approach, it already bridges different levels of formality, models nonfunctional requirements, maintains evolutionary consistency, incomplete models and iterative development, and also handles scale and complexity.

The ATRIUM methodology

ATRIUM [Montero & Navarro, 2009] is a methodology for developing interactive systems, considering both functional and nonfunctional requirements in different levels of abstraction, while using ISO/IEC 25010 as one of the inputs for its process. It counts with a supporting tool to aid on handling each proposed model and activity, with a strong focus in goals and scenarios definition, followed by the generation of a proto-architecture within a synthetizing and transformation procedure. Also, it can be iterated over in order to define and refine the different models.

It is an entirely model-based methodology, guiding the concurrent development from system requirements to software architecture, while dealing with quality issues, as they are considered from the very beginning of its application. Also it is defined as standard quality compliance, especially with SQuaRE, providing the advantage of a proper separation of concerns for the system-to-be. It bridges the gap between requirements and software solutions, using jointly interaction and design patterns, while exploiting these last as solutions for their implementation.

Overall, it allows uncovering any inaccuracies in the application architecture, before code is even written. By using the associated tool, each of its models can be easily described and the traceability throughout its elements maintained, allowing for a continued evaluation of its application. Its recent work continues, more directed to the architectural side on design decisions and anti-patterns.

The STREAM process

The STREAM process originates from an initial approach based on model transformations [Lucena *et al.*, 2009] to generate architectural models from requirements models, where the source and target languages are respectively the i* modeling language and the Acme architectural description language (ADL). It counts on activities as the analysis of internal elements, and the application of horizontal and vertical rules.

Later, this approach to derive architectural structural specifications from system goals had the added development of an activity for selection of an architectural design solution to better achieve nonfunctional criteria, with the possibility to refine an architecture inspired by architectural patterns [Castro *et al.*, 2012]. Also, it presents important work on heuristics which always require experience and know-how from the analysts involved, but it has no current tool support, although an integration with the iStarTool is currently planned.

Associated to this base approach there were a number of proposed extensions, namely the STREAM-ADD, supporting the documentation of architectural design decisions, the F-STREAM, presenting a more flexible and systematic process to derive architectural models from requirements, and the STREAM-AP, devoted to improve the choice of architectural patterns from nonfunctional requirements. Besides these, there is a wealth of other studies surrounding the research community associated to this approach.

The 4SRS method

The 4SRS method [Machado *et al.*, 2006], integrated in a V-Model approach [Ferreira *et al.*, 2014], supports and guides the design of information systems architectures. By successive models derivation based on domain specific needs, it promotes the alignment and traceability between the logical architecture and the requirements supporting models. It begins in a domain-specific perspective at a very high level in the chosen domain and ends with a technological view of the system, represented by a context for product design. The generated models and the alignment between both problem and solution specific domains, as well as the inherent traceability, are represented by a V-Model.

The heart of the V-Model comprises a set of processes with functional requirements handling in a traditional use-case tree, where its leaves serve as the input for the 4SRS method to iterate and derive a logical architecture, built on architectural elements of the future system to-be. As previously referred, since all transformations are model-based, each elicited process aligns directly with one or more elements in the derived logical architecture. These, in turn, can be traced back to their originating requirement(s). The use of an UML-based notation is believed to be more adequate for relating different types of information by leveraging the business requirements and transporting them to implementation phase.

4.2.2 Comparison Framework

The number of proposed solutions for the transformation between requirements and architecture is wide and diverse, either in the way they try to answer the different challenges involved as in the different contexts where they are developed and applied. The need to assess and compare the different approaches, justifying each decision and the origin of the proposed issues, led to a couple of noticeable studies. These focused either in the comparison of different perspectives with a criteria checklist [Galster *et al.*, 2009], mainly composed of a boolean classification of their identified features, and in the systematic review of transformation approaches [Yue *et al.*, 2011a], presenting a taxonomy on its constituents in order to present a conceptual framework for their analysis and comparison.

These previously published results already considered the central issues of requirements elicitation techniques, artifacts on requirements variability and candidate architectures, heuristics and human intervention, iterations and traceability, abstraction and views, and research maturity and quality. Also, in what regards the problem of assessing or comparing approaches in this domain, we too opt for analyzing and evaluating each aspect of the reviewed approaches, in opposite to trying to analyzing the final results of the different methods, as this later is particularly challenging to realize in practice. Following on these studies and also on the earlier analyzed solutions from our four selected approaches, which focus on model-based proposals inspired in the Twin Peaks model, we then propose a framework for assessing model-based approaches on the transformation between requirements and architecture.

In order to assess and compare the different approaches, our proposed twofold framework covers the diverse issues within the transformation from business requirements to software architectures is needed. First, it centers on the issues that represent the core of a model-based transformation approach from requirements to architecture, counting with the representation models of both requirement and architecture elements, as well as the transformation process itself (Table 4.1). Then, in a second view, the issues around traceability, heuristics use and evaluation of each proposed solution are presented (Table 4.2).

Representation models and transformation process

As a high-level view on the transformation method, we first envision it in a black-box style, focusing on what goes in and what comes out. The first issue to analyze is the elicited requirements representation model, where the handling of functional and nonfunctional requirements is transversal to all studies and proposals, except for the 4SRS which only handles functional requirements.

While STREAM is based on the widely used i* language and the 4SRS in UML based standards, ATRIUM uses a mixture of both UML standards, regarding the functional side, and KAOS and NFR for the nonfunctional aspects. The CBSP technique defines its own metamodel for the representation of the base requirements, allowing to identify and isolate nonfunctional requirements at the system level (SP, system properties) and architectural-element level (CP and BP, component and bus properties respectively).

Considering the representation model of the architectural side, the common deliverable is the presentation of an architecture model. Correspondingly in CBSP, the same own-defined metamodel is used to represent the obtained architectural building blocks required to architect a given system, as this method only identifies architecturally relevant information. ATRIUM and 4SRS both use architectural elements based on UML, following OMG-related specification, while STREAM uses the also well-known Acme language. All these three approaches present different proposals, but all highly-modeled and standard-oriented solutions.

In what regards the transformation method itself, it is usually classified as rule, ontological or pattern-based, or even as an identity transformation. In the four selected approaches, most rely on architectural styles or patterns, but there is also a combination of rule and pattern-based, one pure rule-based, and none recurring to ontological-based or identity transformations. Additionally, all of these three more recent proposals work with standard model-based notations, counting with QVT (ATRIUM), ATL (STREAM, this one still in development) and UML supported transformations. The CBSP solution stands a little aside with its close-defined, proprietary process, although it is well defined and modeled. All this information is summarized below in Table 4.1.

Table 4.1: Comparison framework for representation models and transformation method

	RE model (F + NF)	Transformation method / notation	Architectural model
CBSP	CBSP metamodel	Architectural styles / CBSP process	CBSP metamodel
ATRIUM	UML Scenarios, KAOS and NFR	Architectural style and patterns / QVT	Architectural elements (UML)
STREAM	i*	Horiz./Vert. rules and Arch. Patterns / ATL	Acme
4SRS	Processes (Use-Cases)	4 steps rule set / UML	Architectural elements (UML)

Transformation and evaluation

Looking inside the transformation 'black-box', the approaches are normally organized in a series of steps depicted in a workflow-style graph and then described in a step-by-step fashion. Hopefully, they also depict the work products involved, as in the case of ATRIUM and STREAM, and preferably the method is specified and tailor-ready in a standard model-based language like SPEM. Although not in a so standard fashion, CBSP uses ETVX (Entry, Task, Verification, and eXit) to document all its steps in a detailed and clear way.

Regarding other issues related to the transformation method and the evaluation of the obtained architecture, according to the issues raised in past research and in the four selected approaches, we summarize the results in Table 4.2. These issues are related to the desired iterative and traceability support that any proposal following the Twin Peaks model should support, and also the definition of heuristics and a supporting tool to aid practitioners in applying the transformation process, due to their human intensive nature. Lastly, evaluation issues as the execution of a case study or a similar demonstration, as also the existence of a final review on the resulting architecture, are also checked for.

Table 4.2: Comparison framework for transformation and evaluation issues

	Iterative/Traceable	Heuristics	Tool	Case Study	Architecture evaluation
CBSP	Yes / Yes	Architectural styles	Rational Rose	Cargo Router	No
ATRIUM	Yes / Yes	PRISMA models	MORPHEUS	Teach Mover	No
STREAM	No / Yes	Software Engineer	iStarTool *(dev.)*	BTW-UFPE	Suite of metrics
4SRS	Yes / Yes	BP-Analyst / Architect	Spreadsheets	ISOFIN	ARID

All approaches support traceability between the requirements and architecture elements, accordingly to the Twin Peaks vision, nevertheless, regarding iteration (as in the example depicted in Fig. 4.1 for the 4SRS associated V-Model), the STREAM process does not seem to be adequate for it. All the other approaches support increasingly detailed iterations to further refine the requirements and architecture elements. This also means that STREAM can not initiate from the architectural side, being a purely transformation from requirements to architecture [Cleland-Huang, Hanmer, Supakkul, & Mirakhorli, 2013].

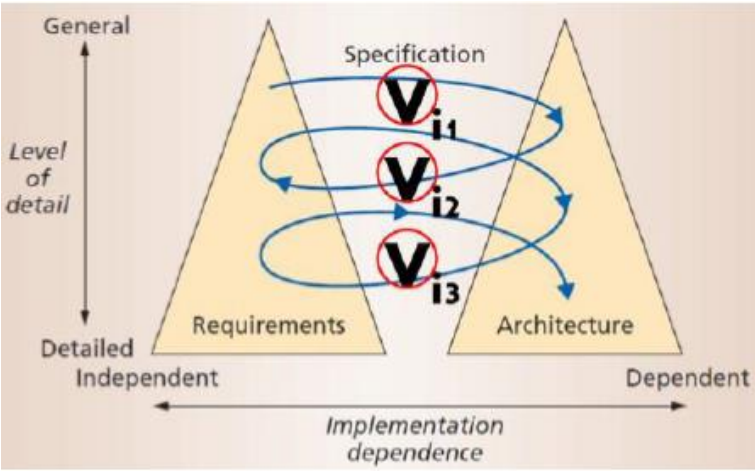


Fig. 4.1: V-Model integration with the Twin Peaks model

Heuristics are also an essential part of any transformation process, as human participation is always required (although tool support can accompany its application). While CBSP and ATRIUM impose stricter rules, the STREAM and 4SRS approaches allow for more human intervention, namely from the software engineer and the business-process analyst/architect, respectively.

All the analyzed approaches involve a Case Study in the evaluation of their proposals and all adhere to tool usage, mostly counting with open access development platforms (the STREAM solution is still in an initial development phase). In this last issue, the exception is on the 4SRS method which relies mostly on the use of spreadsheets to support its execution.

As previously referred, the plain evaluation of a transformation proposal is no longer enough, an added evaluation of the quality of the resulting architecture is also important nowadays. The STREAM approach already performs an evaluation based on a previously defined suite of metrics, but plans to advance for deeper evaluations. Regarding the 4SRS, the Active Reviews for Intermediate Designs (ARID) [Clements, 2000] is the method of choice used for the evaluation of the ISOFIN logical architecture, as described in one of its most relevant recent projects [Ferreira *et al.*, 2014]. The other two analyzed approaches do not refer any quality evaluation on their corresponding resulting architecture.

4.2.3 Discussion

Diverse approaches are currently available, using different strategies to handle the connection between requirements and architecture according to each intervention setting. Although structurally alike, each approach has its proper insights. The 4SRS tries to follow on the same structure of the other similar approaches, but also presenting its added values, as it is associated with an higher number and more recent set of research publications, representing a continued research effort only followed by STREAM, but at a fair distance (the other two did not have significate follow-up research), and supports iteration in the transformation process, being able to initiate it from either the requirements or architectural sides, two important features that STREAM does not comply with.

Although there are several other similar approaches in the research universe, these four we selected were the ones that comply with most of the issues being analyzed and that can set the trends for future research in this area. The usage of model-based languages and tools as well as standard-based reference models is increasing, being almost omnipresent in recent research proposals.

Questions as the inclusion of nonfunctional requirements, added to the traditional functional requirements, and increased balance between hard transformation rules and loose user heuristics are also ubiquitous. Although several heuristics have been proposed for each existing process, being more or less creativity and knowledgeability-dependable, most times the final decision is left to the responsibility and decision of the analysts. Even so, the need to further refine existing heuristics and develop new ones is ever-present.

Case studies remain the preferred way to evaluate and validate any proposal, but architecture evaluation is also on the mind of researchers already. Even so, further interrelated and innovative evaluations are always recommended, especially when dealing with quality issues and complex scenarios.

Relative to alignment and traceability consistency, all proposals seem full-proof as all its steps and elements are modeled. Nevertheless, there is still some work to do, relative to a quicker access in obtaining related elements, especially involving manual tasks performed by analysts, as not all the procedures are tool supported. So, additional tool support development is an interesting asset for any of the proposals.

Open questions to be dealt with are also in the poor formalism among the different approaches, with the use of different terms to classify the transformation (approach, method, methodology, technique and process) and also for the act itself (transformation, transition or derivation). Another question refers to widen the scope of the approaches from modeled requirements to business ones and from system architectures to ones closer to software specification.

The Twin Peaks model has set the foundations for much of the research around the transformation from requirements engineering to information system architectures. In the meantime, research has advanced those basis and several new trends, in diverse research directions, have been proposed, so there is a need to classify and compare them within their constituents.

The proposed framework allows assessing our groups' proposal, while framing and comparing it, relatively to other similar and relevant model-based approaches in this domain. By performing a review on their transformation issues from requirements to architectures, it covers each of the previously identified research issues and then classifies each of the considered approaches accordingly.

Some ongoing projects within our research group (SEMAG), affiliated to the ALGORITMI research center, already integrate the V-process method, including the 4SRS method. As this analysis makes us more aware of its characteristics, strengths and limitations, our work intends to evolve these methods by aligning the existing metamodels and integrating others, while improving current features and bringing new ones to strengthen those solutions. Also, according to our principles, we intend to walk on model-based solutions, with extensive use of metamodels, particularly on the OMG specifications previously identified, and in continued method engineering.

4.3 DESIGN OF THE PGR APPROACH

As a first part of our study we frame the elicitation and management of requirements inside a software development methodology. Between the high-level, complex, RUP and the agile, lightweight, Scrum, the V-Model process-level approach with its integrated 4SRS method positions itself inside the agile models. It fits particularly within the RUP methodology through the phases of inception and elaboration, traversing the disciplines of business modeling, requirements and analysis & design (Fig. 4.2).

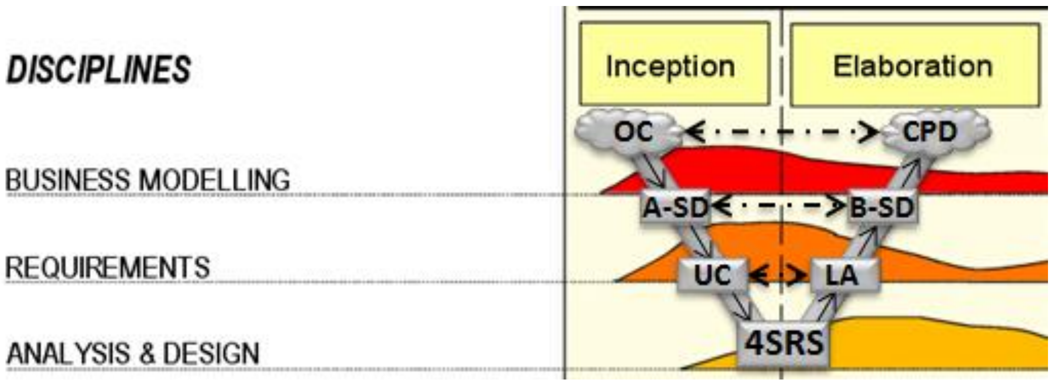


Fig. 4.2: Framing the V-Model method inside the RUP methodology

From inception to elaboration, the initial organizational configurations (OC) and derived context for product design (CPD) focus on the vision of the organization in which the system will be deployed, including the vision and business case definitions, as well as the main use-cases. Accordingly, the A-type sequence diagrams (A-SD) and the B-type ones (B-SD) represent the high-level view for a set of requirements work products that scope the system to be built, while the use-cases (UC) and the Logical Architecture (LA) provide the detailed requirements for what the system must do.

Furthermore, inside the V-Model the 4SRS stands as the nevralgic center for the transformation between requirements and architecture, focusing on the alignment levels between use-cases and logical architectures (Fig. 4.3). It is in charge of delivering a design model which consists of design classes structured into packages and subsystems with well-defined interfaces, representing what will become components in the implementation. All these compose an initial vision showing how the system will be realized, which are according to the related RUP artifacts. As discussed previously, this specific focus allows us to first study the requirements engineering topic, establishing a solid ground from where to connect to the related business model and enterprise architecture topics.

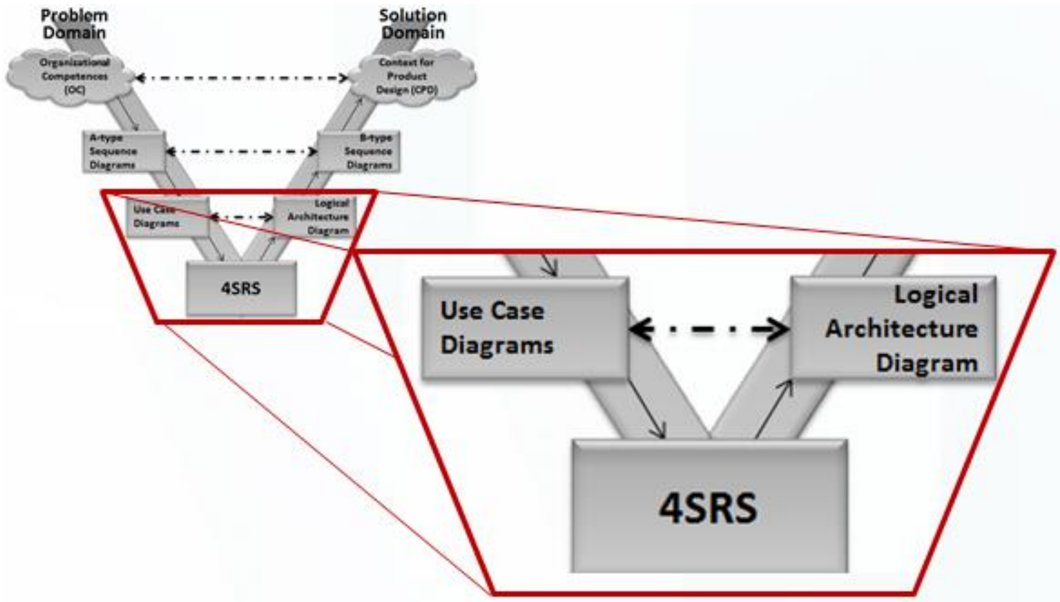


Fig. 4.3: Focus on the transformation alignment between requirements and architecture

Many studies state the importance of the business model and strategy definition in the early specification, and during the evolution, of a software-based information system [Al-Debei & Avison, 2010]. Despite the many methods and techniques available to build them, it is not easy to do it in a clear way, due to a number of factors, especially in ill-defined contexts.

In our current practice, the use of process-oriented approaches as the V-Model [Ferreira *et al.*, 2013], allows eliciting requirements and obtaining a process-view of an information system in such contexts. Results normally consist in scenarios and processes that describe the expected behavior of the organization, with the V-Model using UML sequence diagrams and use-cases, later derived into a logical architecture and product-level system information. Albeit the good results achieved much information is lost by not uncovering the business vision and strategy information artifacts, so handicapping the obtained solution and its future evolution. As seen in the framework presented in the previous section, this is mainly due to the lack of treatment for the nonfunctional requirements.

To overcome this insufficiency some studies have been researching on the importance of integrating scenarios and business information, whether by deriving requirements models from organizational models by means of goal modeling [González & Díaz, 2007] or by using organizational models to improve use-cases development [Santander & Castro, 2002]. As usual, these studies follow a traditional development sequence, deriving from business and requirements elicitation to process definition and detail.

On the other hand, opposite approaches that focus on the discovery of goals from scenarios, to directly help in the requirements engineering activity [Rolland, Souveyet, & Achour, 1998] or in the process of inferring formal specifications of goals and requirements from scenario descriptions [Lamsweerde & Willemet, 1998], have also been explored for some time. In a more recent example, the validation of mappings that describe the realization of goals by activities (supported by models), while considering tasks as goal-oriented requirements, was proposed [Gröner *et al.*, 2012].

Relying on model driven methodologies to better handle requirements, the transformation from business to process can be performed backwardly, but as process information is not as rich as business information, other methods and techniques are required to help in filling some holes. So, we propose to combine the advantages of visual modeling with a well-established process, following the business modeling guidelines from the Rational Unified Process (RUP) and the business plans representation of the Business Motivation Model (BMM), in order to complete the business model and strategy definition for the referred information system.

Despite the promising results already obtained regarding the V-Model process modeling and alignment issues, there is still much room to improve in its associated requirements elicitation and traceability management. Relevant contributions are expected from the strategy and business model topics, so, we propose to extend it with a BMM representation, guided by a RUP-based backward transformation from process to business.

Both RUP and BMM rely heavily on modeling and are standard-oriented, as modeling is crucial for the continuous alignment between business and IS, and the improved traceability between both. With relation to BMM, it is already full modeled and UML compliant, while regarding RUP, there is already a solution modeling its textual artifact, the document for business vision, in UML [Cooper, Abraham, Unnithan, Chung, & Courtney, 2006], but needs some updates. On the other hand, SoaML, with its placeholder connection to BMM will allow expanding this work to the architectural side.

Relating to methods and techniques for eliciting goals and rules, notwithstanding other elicitation methods and techniques as i* [Yu, 1997] or KAOS [Dardenne *et al.*, 1993], this choice for the associated combination of checklists and guidelines from RUP, and in the business plans representation of BMM, is due to the more complete and business oriented side of RUP and BMM. This side helps in defining the business requirements specification for business modeling, while promoting the information systems and business alignment questions that are comprised in process-oriented approaches.

4.3.1 The PGR metamodel

According to these previous considerations, our approach for inferring business and strategy information from scenarios and process-like diagrams uses guidance from the RUP Business

Modeling discipline, backwardly performed, and the business plans modeling representation capability of BMM.

Analyzing the RUP and BMM metamodels with regard to the three concepts of Business Use-Case/Process, Business Goal and Business Rule, we first build a unified relation between them. In RUP, Goals and Use-Cases are directly connected, while Rules and Use-Cases are connected through an Entity. In relation to BMM, Rules are directly connected to Processes (Use-Cases), while Goals are connected to Processes through an Organization Unit. As at this phase we are not handling any concept regarding Entities nor Organization Units, these are left out of the proposed relationship.

The final diagram to unify those three concepts is depicted in Fig. 4.4. Also, there is the predefined BMM direct connection to SoaML (a placeholder with the capability for connecting SoaML to BMM models is provided to capture how services solutions realize business motivation). We see this simplified metamodel as a model's model that serves for explanation and definition of relationships among the various components of the applied model itself.

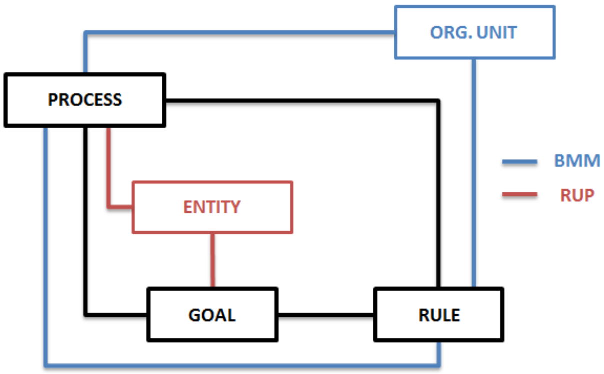


Fig. 4.4: RUP and BMM unified relation for use-cases, goals and rules

Notwithstanding, this plain representation is insufficient for today's demanding needs, as well as for a sound and detailed business model. So, expanding the contribution from BMM, in a more business-like representation, we detail the previous schematics with Business Rule and Business Goal specific elements, maintaining the Business Use-Case, Business Rule and Business Goal relation (Fig. 4.5).

The details of the metamodel allow for the relation between these elements (represented in the B side of Fig. 4.5):

- Process (Business Use-Case), in the form of a more or less detailed Business Process structure, where all of those Processes, plus the Mission and Vision statements, connect to the Organization element;
- Goal (Business Goal), a Goal detailed by Objective and connected to the Vision statement;
- Rule (Business Rule), representing Business Policy and Strategy, these respectively detailed by Business Rule and Tactic, with Strategy linking directly to the Mission statement.

4.3.2 The PGR method

With this representation in mind, the first step will be to perform the elicitation of goals (Desired Results) and rules (Courses of Action and Directives) for each root use-case in the V-Model structure, elements represented in the A side of Fig. 4.5. This elicitation follows backwardly upon the RUP guidelines, initiating with the V-Model's sequence diagrams and use-cases scenarios information, in order to obtain their associated goals and rules. The concepts/associations represented in Fig. 4.5, in both sides A and B, derive directly from the BMM specification diagrams previously presented in Subsection 3.3.1 and further detailed in the BMM specification[40], while the elicitation of nonfunctional requirements, namely goals and rules, has also been previously described in Subsection 2.2.3.

According to the RUP guidelines there should be at least one goal for each use-case, with their classification being assisted by the use of the BSC technique [Kaplan & Norton, 1996]. Regarding rules, the same process applies, but these being categorized by either constraint or derivation types. The BMM guidelines regarding the elicitation of goals and rules are also of great importance in this phase, especially to be used as the base heuristics by the less experienced business-process analysts.

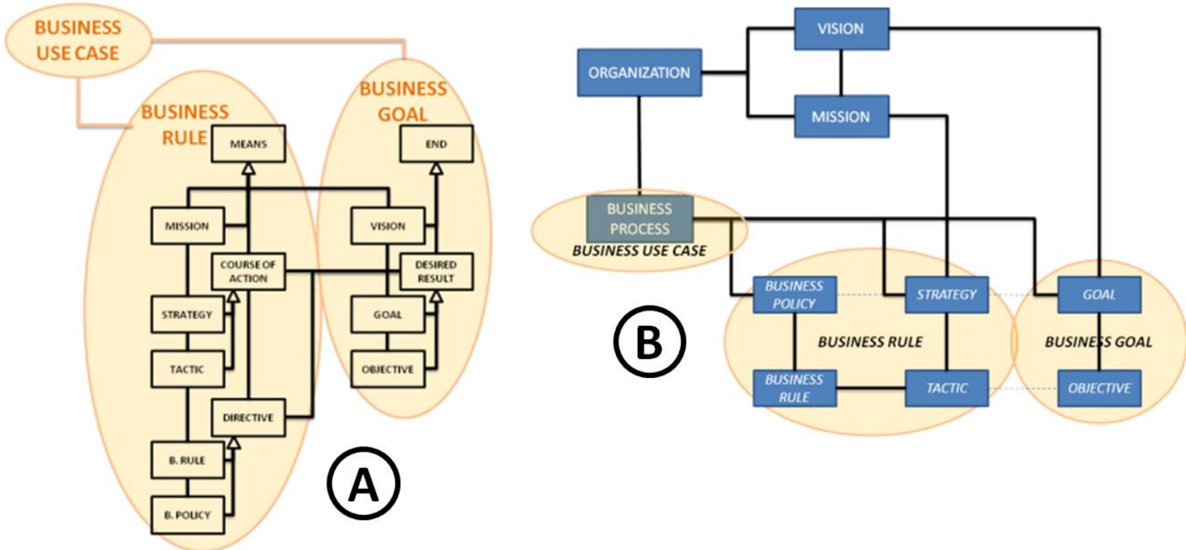


Fig. 4.5: BMM business rule/business goal detail and the simplified business model

In a second step and going deeper into the V-Model leaf use-cases, the refinement of goals, including objectives at the lower leaves, and rules, or associated strategies, tactics and policies, will raise conflicts, new associated goals or rules, and reflexive changes upward [Lamsweerde, 2008]. This process initiates with a top-down approach, but soon intertwines with bottom-up techniques, using How and Why questions respectively to go up and down the use-case/goal/rule tree (Fig. 4.6). The entire process should take several iterations until all stakeholders involved in this method are satisfied with the results.

[40] http://www.omg.org/spec/BMM/1.3/

Additional associated business strategy techniques and methods (SWOT and Porter's competitive analysis, among others), alongside the previous referred BMM guidelines, as well as all the collectable stakeholders knowledge contributes and accompanying specialists' heuristics [Gigerenzer & Gaissmaier, 2011] are welcome in this step to aid in building a well-balanced, agile and realistic goal/rule hierarchy.

As steps 1, 2 and 3 (Fig. 4.6) fill up the PGR structure, we are ready to, in a fourth step, either to generate a business plan, following the work on the top-level use-cases, or to derive a candidate architecture, using the bottom-level use-cases to feed the 4SRS method.

In the first case that could be done through abstraction of the goals and rules associated to the V-Model top-level root use-cases, following Why questions, and aided by the BSC classification technique and information gathered from the RUP Business Vision guidelines. If pertinent, additional intermediate rules and goals can be defined to strengthen the connection between the high Vision and Mission statements and the intermediate root use-cases goals and rules.

Regarding the second case, the bottom-level leaf use-cases are ready to integrate the remainder of the V-Model process, with strong implications to the heart of it, the 4SRS. The entire dynamic of the 4SRS can be strengthened with the inclusion of the goal and rule elements, aiding and supporting the decisions taken within. These two topics are addressed in the next chapters.

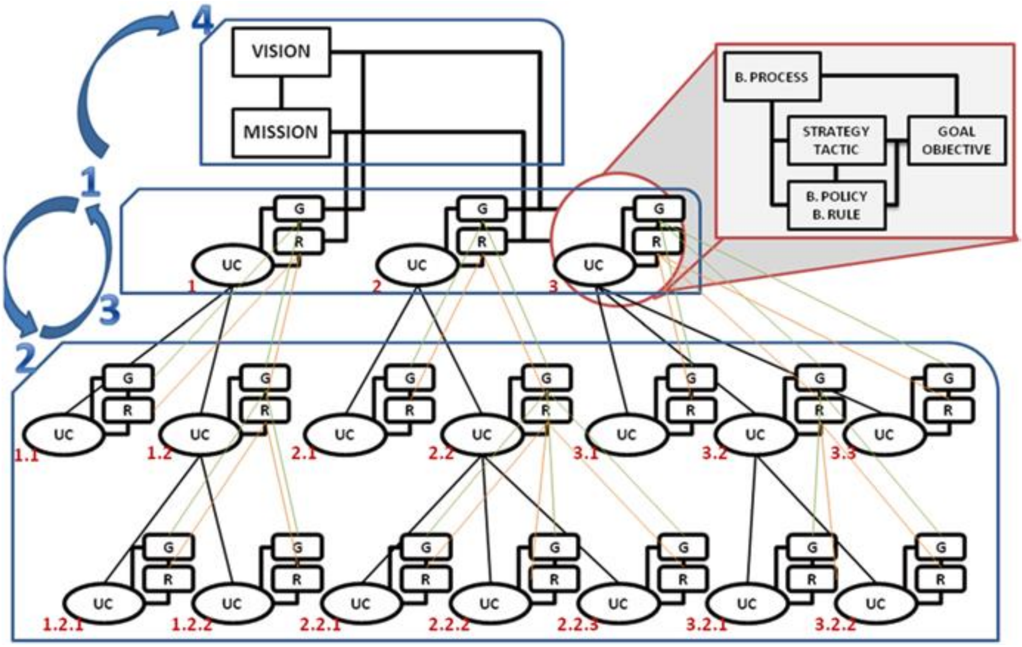


Fig. 4.6: Use-case tree with added goals and rules relations

This approach is not exhaustive nor are all its steps mandatory for every project. Its aim is to stand as a flexible set of steps and techniques, for, in conjunction with the V-Model process, to be tailored to each project and according to each situation. Some might go deeper in the goals and rules derivation, others can emphasize more the high-order statements, so taking different paths and making distinct choices.

So, the decision on the number of iterations to perform is influenced by the time and resources available, the intended depth and detail to obtain, and also on the complexity of techniques to use for elicitation. These last can whether rely on simple heuristics or recur to the users' most preferred advanced methods, all entirely user-free to decide.

In order to deliver a tailorable solution for the PGR method ready to be adapted and used in a project, we propose the design of an activity (using the SPEM specification). With this definition at hand this method counts on a specific set of tasks, work products and roles. It is able to be performed by a business-process analyst, while organized in three iterations of elicitation, refining and revision of both business goals and rules from business processes (Fig. 4.7).

The activity named "Inferring business and strategy information from scenarios and process-like diagrams" relies on the following tasks:

1.a) Elicit business goal;

1.b) Elicit business rules;

2.a) Refine business goal;

2.b) Refine business rule;

3.a) Revise business goal;

3.b) Revise business rule.

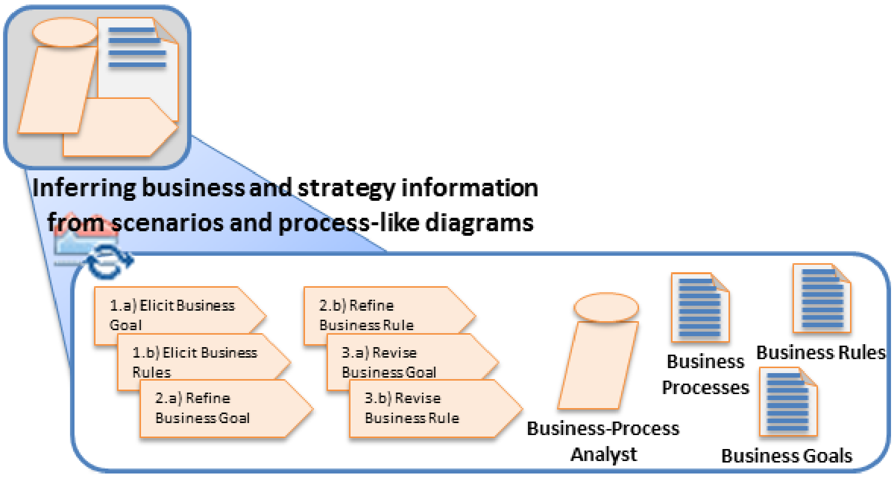


Fig. 4.7: PGR method associated tasks, work products and roles

Which generate or use the following work products:

- Business processes;
- Business goals;
- Business rules.

As referred, all the responsibility to perform the tasks and handle the work products in this activity is on the hands of the role of a business-process analyst.

Due to this specification the activity can be tailored in a SPEM process, operationalizing it according to the users' intentions. In this case we present an example (Fig. 4.8) of using the designed activity in a process to elicit the related business goals and rules from the initial associated use-cases, executed iteratively, followed by the refining and revising of these same and other associated goals and rules, organized in two nested, iteratively cycles. The outputs of this activity are the business goals and rules work products.

4.3.3 Discussion

The described guidelines can be applied to any information system project by starting from the already existing V-Model Sequence Diagrams (A-type) and Use-Cases, reflecting those to RUP's BUC, and then eliciting Business Goals and Rules for each one of them. Next, the iteration to infer higher and derive lower goals and rules, allows aggregating and hierarchizing them.

Notwithstanding, the RUP guidelines are not exhaustive, neither for the business goals and rules elicitation nor for the business vision statements, being sometimes even dissonant, with some of the decisions having to rely on stakeholder contributions or specialists' heuristics. On the other hand, the simplified BMM modeling representation proposed supports all the information uncovered from the V-Model and generated from the RUP guidelines, contributing also with a few pertinent suggestions for elements elicitation and model completion.

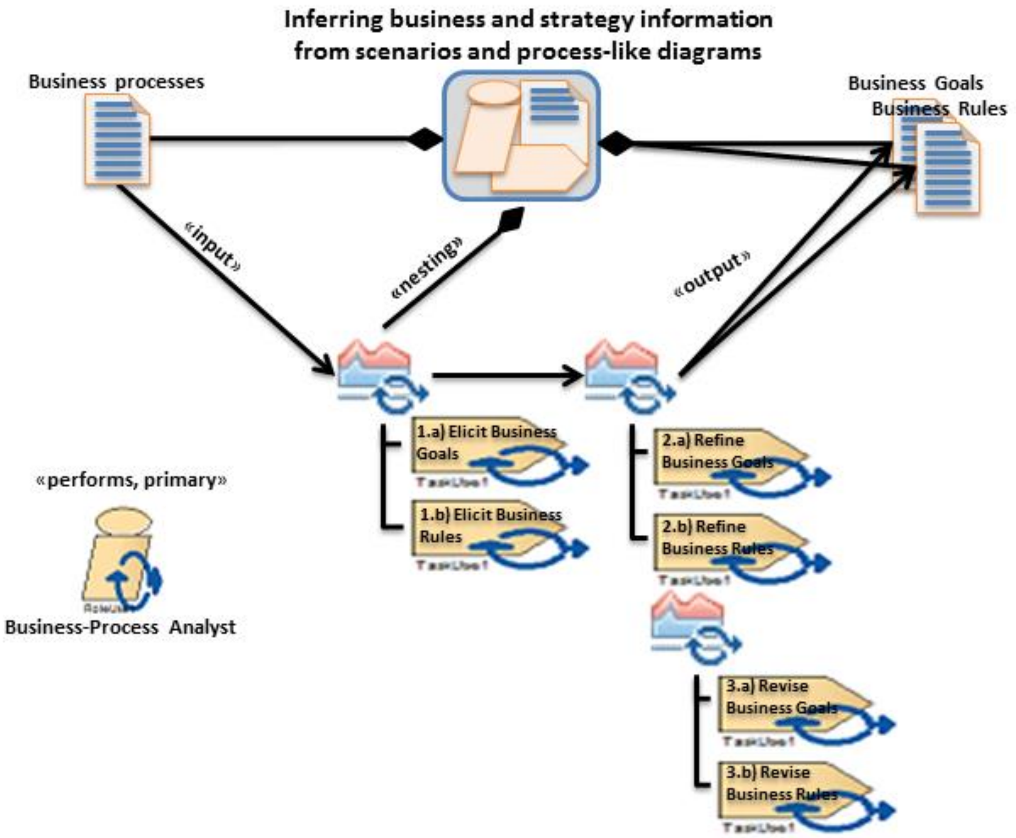


Fig. 4.8: PGR method process tasks, work products and roles

The associated activity specification straightens the path for the development of a tool-based support, benefiting a larger community to be able to perform the steps described, while helping to avoid the error-prone situations of manually mapping from RUP to BMM. It also serves as a helpful guidance, while setting a similar language for different stakeholders with the added metamodel association from RUP to BMM, which strengthens our proposed solution.

4.4 TOOL SUPPORT: THE PGR MODELER

According to the previously analyzed tools (Section 3.4), our choice for the development of an agile prototype available through an aPaaS solution was the OutSystems Studio. Besides being a Portuguese company, relying on some of our former colleagues for liaison support, its ease of development, short learning curve and breadth of functionalities justified our picking.

As its fellow competitors, the OutSystems platform provides a lifetime free development environment for small projects where all the core capabilities are available. The only demand for keeping it active is to access it regularly otherwise it will fall asleep and be kept in backup. It relies on a quite comprehensive training area and extensive developer's forums for support, and also a community forge for open source components and application sharing. Besides the online management area for each user, counting with applications, users & roles (feature not available for free accounts), environments (1 server with a 2GB database for a free account) and analytics (feature not available for free accounts), a desktop tool for development, the OutSystems Studio (currently on its 9.1 version), is also available.

Visual development environment counts with four layers: Processes for handling the high-level business processes; Interface to define the navigation flows of the user interfaces; Logic for implementing specific business rules; and the Data layer with the application data module. It allows for quickly creating applications in a responsive web design that adapts to the form factor of the device where the user accesses it, following a WYSIWG edition, with the platform ensuring consistency and validity, automatically healing the application where possible. With one click it builds and deploys the application making it readily available anywhere in any browser.

According to our needs, we will be using mostly the Data and Interface layers, since due to the relatively low-size and complexity of the intended prototypes there will be no need for defining high-level business processes or broad business rules. Next we detail the relevant decisions made in these two layers for the PGR Modeler prototype and additionally browse the working environment with some screenshots of the application.

4.4.1 Entities Structure

In order to implement an application to support the referred metamodel, a set of corresponding entities were created, along with the necessary adaptations. The list of entities is created in the Data layer of the OutSystems Studio, alongside the attributes of each entity and the general CRUD (Create, Read, Update, Delete) operations to perform with them (Fig. 4.9), one of the many scaffolding assistances offered by this development environment. Also included in this layer is the possibility to sketch an entity diagram with the relations between them (Fig. 4.10).

First of all, an entity PRJ was created to allow for the creation and maintenance of several independent PGR for different projects, otherwise only one PGR could ever be created with this application. According to the metamodel, the use-cases processes are represented in an entity UC, while the goals and rules dimensions have been aggregated in one single entity, GR. This organization of the goals and rules dimensions in one entity, which in our case will be the six BMM dimensions (Goal, Objective, Strategy, Tactic, Business Policy and Business Rule), require the existence of another entity to represent the goals and rules statements associated to each pair use-case, goal or rule dimension.

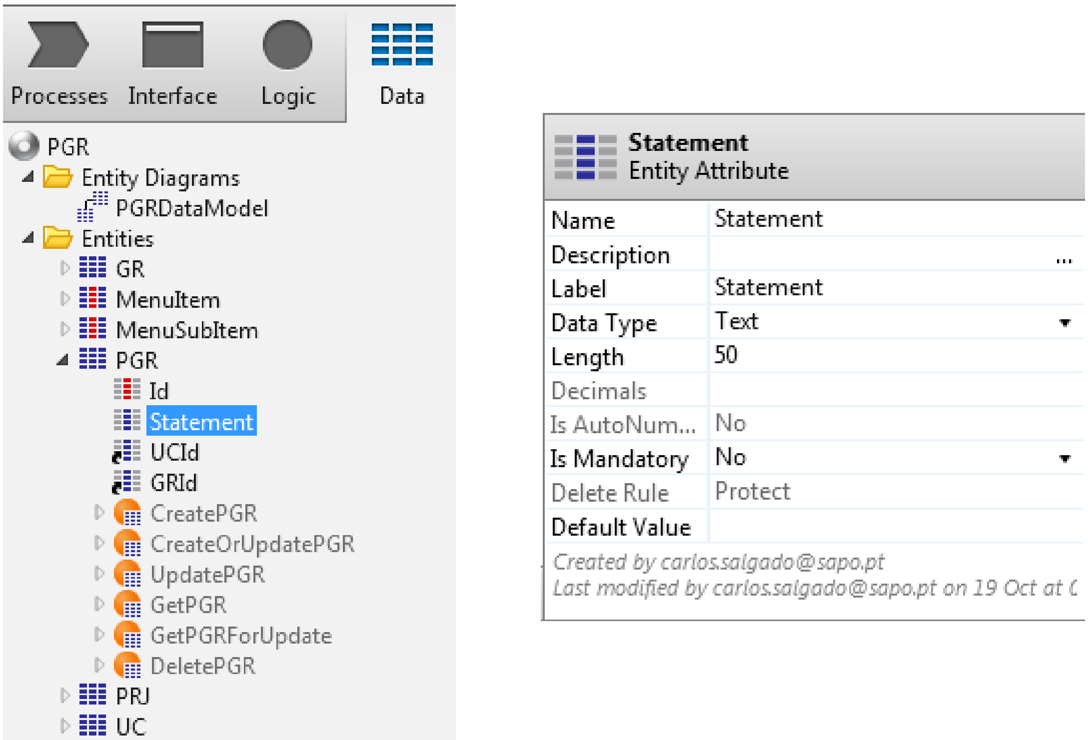


Fig. 4.9: Data Layer with the list of entities and detail for the Statement attribute of table PGR

As can be seen in Fig. 4.9, the PGR entity possesses external connections to the UC and GR entities (through the UCId and GRId attributes), and a descriptive attribute Statement that allows to represent a goal or rule statement, associated to a specific use-case. Some examples for this situation are provided later on in this section. Moreover, further examples of the use of this tool are listed in appendices at the end of this document.

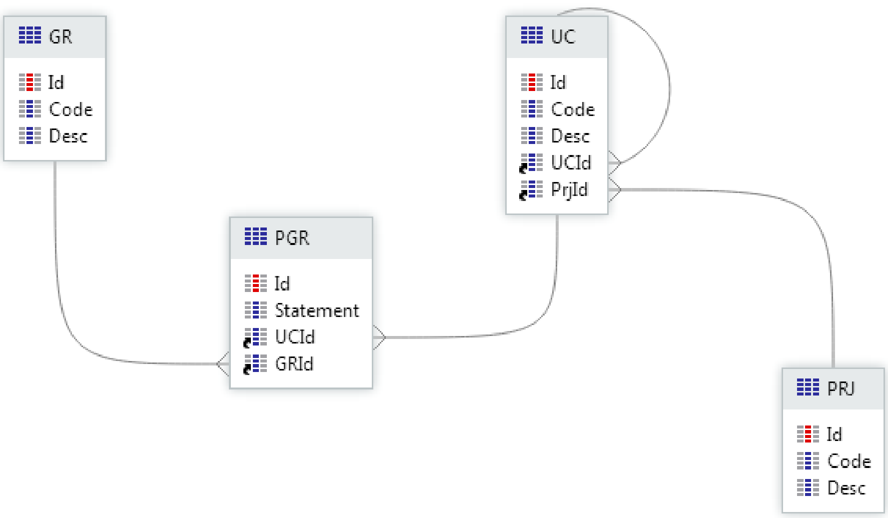


Fig. 4.10: Entities Diagram

The complete diagram for the PGR data model is presented in Fig. 4.10, where all the relations and dependencies between the different entities can be easily visualized. Note for the possibility of the use-case entity to reference itself, which allows for the detailing of a use-case in several related sub-use-cases (Fig. 4.6), with the leaf use-case referencing its parent use-case. This stresses for two exceptions, the top use-cases do not need to reference a use-case and a specific use-case should not refer itself.

4.4.2 Screen Flows

After the entire entities structure is ready, it is time to design the interface screens and the associated flows connecting them. Besides that, local entry variables can be added and associated events defined for each screen. The most common of the events, present in every screen, is the Preparation, where any specific business or technical logic can be encapsulated to prepare data for the screen design. This is all possible through the Interface layer (left side of Fig. 4.11). An example of a code 'flowchart' in the Preparation event for the UC screen is shown in the right side of Fig. 4.11.

In the MainFlow, with an initial Home screen by default, the developer can add the desired screens for the application. Again here there is an interesting scaffolding assistance in quickly creating screens to handle a chosen entity, just by dragging and dropping the entity in the screen canvas space. This generates a typical listing screen for the elements of that entity (Fig. 4.12), with associated filters, attribute sorting and others.

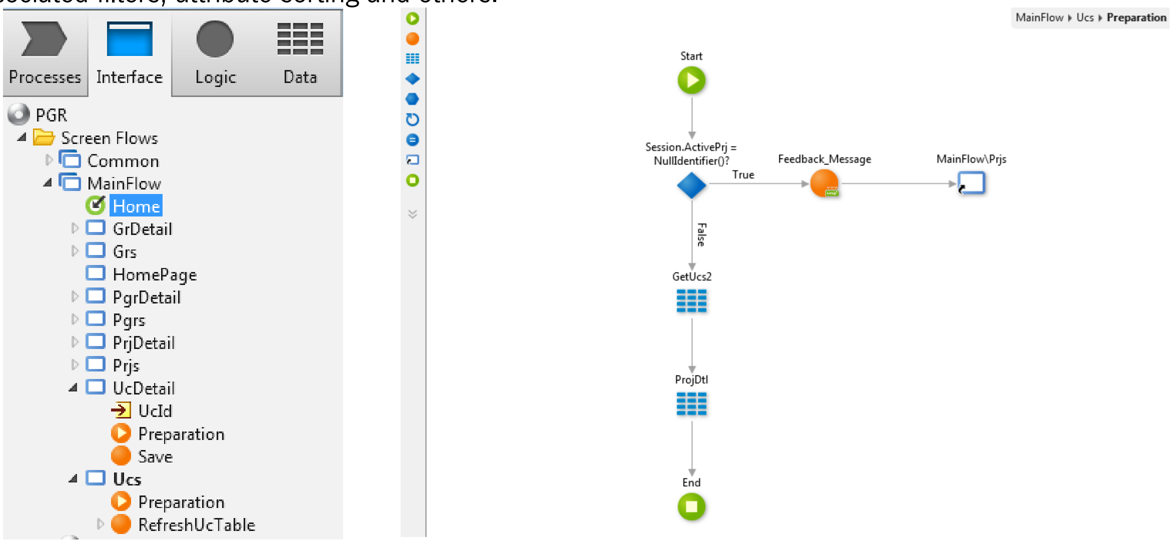


Fig. 4.11: Interface Web Screens, local variables, preparation (with Ucs detail) and events

Moreover, by dragging a second turn the same entity in the canvas, a typical detail editing screen for the elements of that entity is also created automatically, with all the links between the two screens and all the needed validations. This was the case in the four existing entities, with the few needed adaptations done afterwards on top of the pre-generated screens. The set of generated screens is shown in Fig. 4.13, with the associated flow between the screens represented by arrows. The dependency defined between Ucs and Prjs is related with the need to first select an active project in order to gain access to the associated use-cases.

Again referring to the encapsulation of some needed business or technical logic in events, its 'coding' is done in a completely visual way, with the editor making available elements to define one start, and one or more end points, to spark the execution of other events, to build aggregates that fetch data from entities, to control the flow with if and switch questions, to define loops, to set variables and to navigate to other screens (top-left of right-side screen in Fig. 4.11).

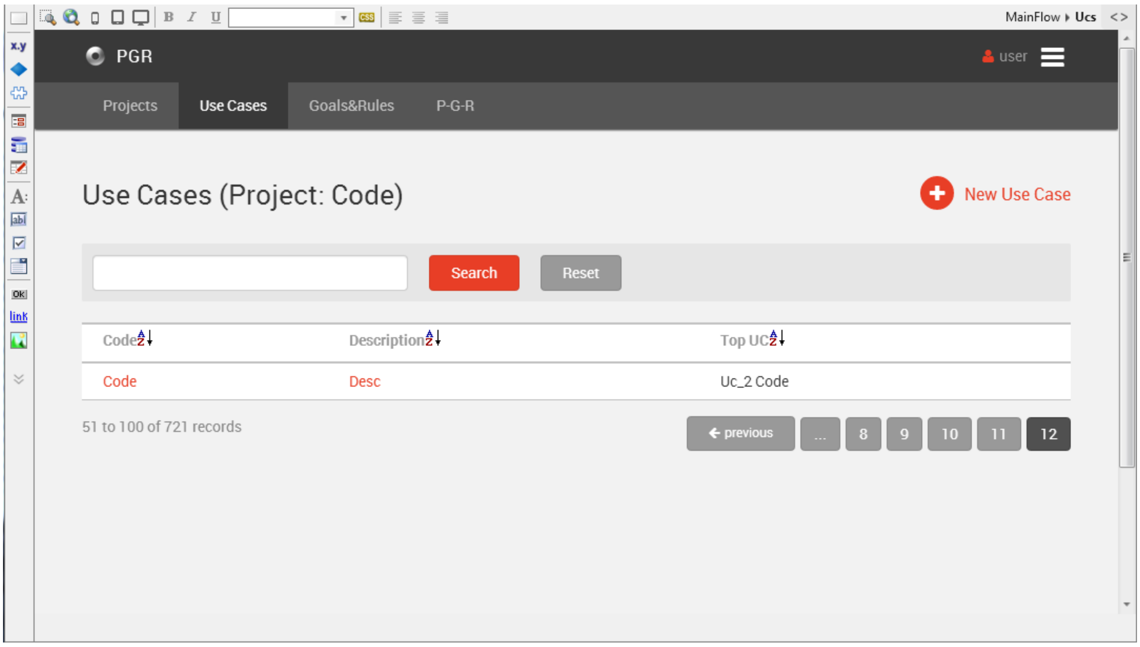


Fig. 4.12: Design mode for Web Screen

4.4.3 Form captions

Finally, after the entities structured, and the screens designed and properly prepared, the application can be generated and deployed in the server. This concluded, it is possible to immediately navigate the application and perform the desired operations. In the case of the PGR Modeler application, the first operation should be to create a new project and set it active. Next, its associated use-cases can be defined, while the goals and rules dimensions (Fig. 4.14) can be defined at any time.

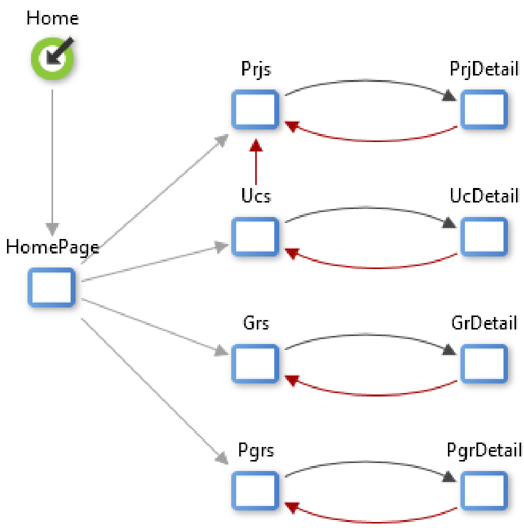


Fig. 4.13: Web Screens flow

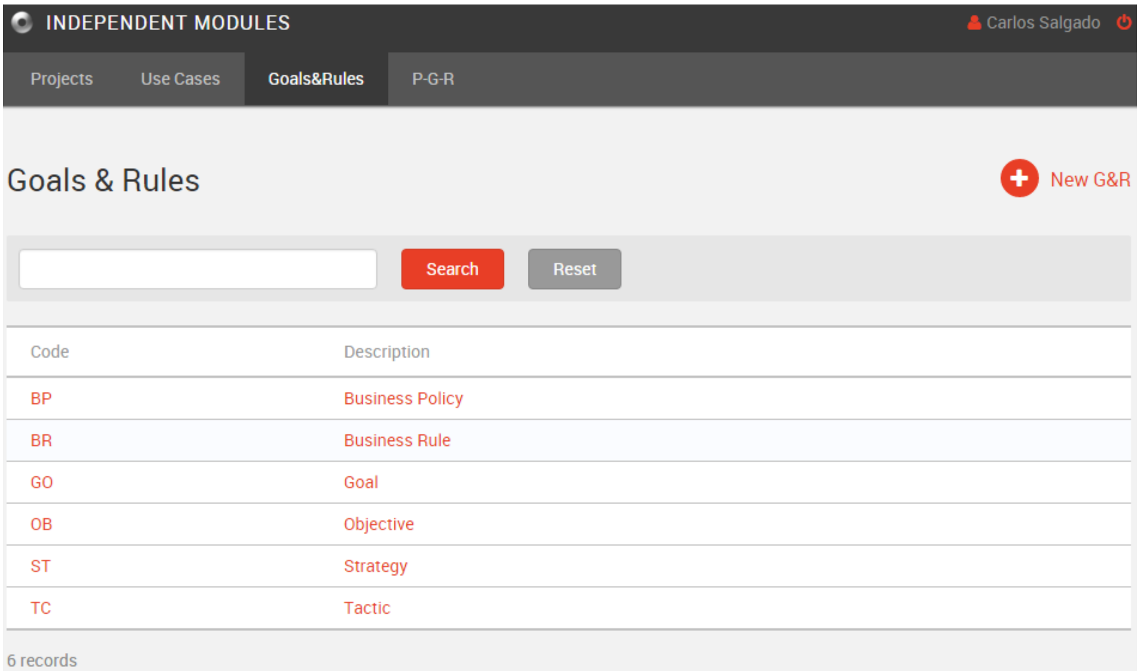


Fig. 4.14: Goals & Rules online listing

The main step of defining PGR statements can be performed thereafter, by repeatedly creating new PGR elements, associated to a specific use-case and a goals/rules dimension (Fig. 4.15).

The complete application can be accessed and freely tested in the following web address:

- https://carlossalgado.outsystemscloud.com/PGR/

This prototype tool completes the contribution to create and handle a model-based solution consisting in structuring top, middle and base-level use-cases complemented with nonfunctional data in the form of associated goals and rules, all organized in a tree-shape format. Two projects were used for testing this prototype tool (AC.pt and AAL4ALL), but with no specific case study to analyze and discuss. Appendix A contains examples of the use of this tool regarding the first of these two projects.

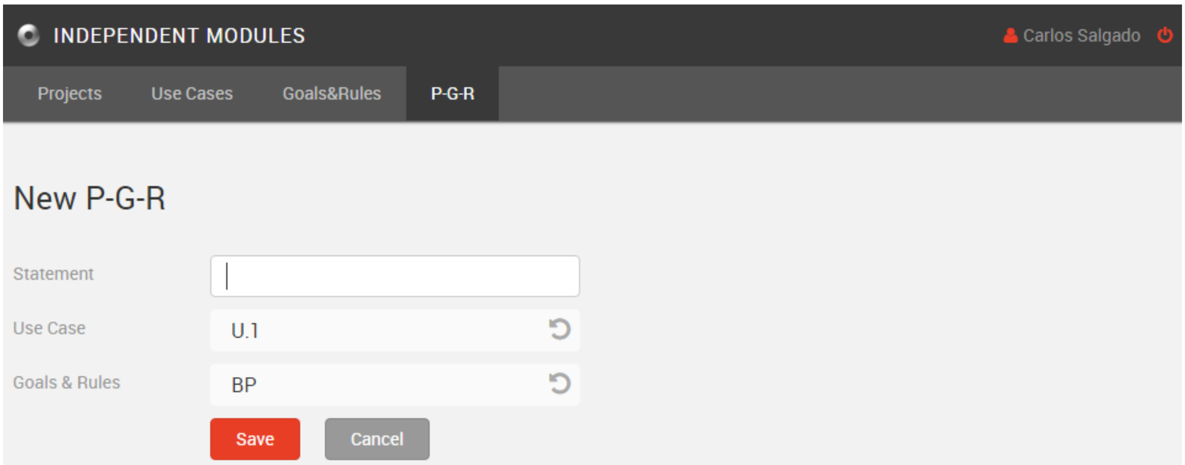


Fig. 4.15: PGR detail online form

4.5 CONCLUSION

The importance on the use of models within the transformation between requirements and architectures is as high as in other crucial engineering task. Although the lack of widespread use and community adherence to the consistent use of models in this topic, a diversity of tentative model-based solutions have been proposed with special relevance to the ones inspired in the Twin Peaks model. The task of requirements elicitation stands as a first step in this transformation where the nonfunctional side, associated to business quality characteristics, has not been receiving as much consideration as the functional, more practical side, of requirements.

Regarding the comparison between a chosen set of methods aimed at the transformation between requirements and architectures, we analyzed their model-based solutions mostly recurring to popular references with iterative, traceable approaches, also counting on founded heuristics, tool support and accompanying case studies. Among them we highlight our research groups' 4SRS method which stands out for its continued evolving research and recent tentative architecture evaluation. Nevertheless, it still has its weaknesses and limitations, some of which we will try to improve along this work. In this first step our work is restricted to handling requirements while the architectural side will be dealt with in chapter 6.

More on the organizational side, as business processes are increasingly becoming the focus of information systems (even as the primary choice for their requirement elicitation) they cannot be disconnected from the organizational environment. Aiming to provide a business and strategic view of an information system, we adapt standard techniques to infer goals and requirements from scenarios and process-like diagrams, mapping backwardly the traditional business to process workflow.

Our proposed PGR approach, composed of a metamodel, crossing functional and nonfunctional requirements, and an associated tailorable method for their elicitation, is inspired in the business plans representation of BMM and in the business modeling guidelines of the RUP methodology. The PGR-based requirements integrate the left-side, descending branch, of the V-Model method, adding goal and rule elements to the early plain use-case, process-view solution, in so answering to current information systems' demanding needs. Accordingly, a tool support that allows for the information system analysts to build the use-cases, goals and rules trees for each individual developed project or organization is presented at the end.

Directly following the work of this chapter, there is the connection to business models directly from the top-level use-cases, relating to the vision and mission statements, and associated strategy of an organization. The BMM business plans stand as a first step to link to other reference models that are able to generate a business model, as consecutively proposed in Chapter 5.

Accordingly, by following the V-Model development, the connection between requirements and architectures can also profit from the use of the PGR-based requirements by taking in consideration the direct link between BMM and SoaML, namely on its business goals and rules placeholder. This demands further improvements in generating adequate SOA logical architectures, starting from the bottom, leaf-level, use-cases that normally feed the 4SRS method. These issues are also handled in Chapter 6.

—— This page intentionally left blank ——

CHAPTER 5

GENERATION OF BUSINESS MODELS FROM PGR-BASED REQUIREMENTS

Summary: Business models are pivotal in building bridges between business and technological worlds, although its associated research is still young and in an immature state. It is possible to develop a business model through reengineering the elicitation of processes, goals and rules for an information system nevertheless, this method involves some complexity. In order to handle these issues, a three-dimensional proposal combining the elicitation of PGR-based requirements, with a focus on BSC perspectives and supported in a metamodel for its representation and a tailorable method for its manipulation, is presented. Furthermore, associated to it there is the development of a tool support within the OutSystems Studio environment, with its entities structure, screen flows and form captions.

CHAPTER TABLE OF CONTENTS

PUBLICATIONS

- Generating a Business Model Canvas through Elicitation of Business Goals and Rules from Process-level Use Cases (**BIR'14, Springer**)
- Generating a Business Model through the Elicitation of Business Goals and Rules within a SPEM approach (**ICIST'14, Springer**)
- A Three-dimensional, Requirements based, Balanced Scorecard Business Model (**ICICS'15, IEEE**)
- Exploring a Three-dimensional, Requirements-based, Balanced Scorecard Business Model: On the Elicitation and Generation of a Business Model Canvas (**CBI'15 [TEE], IEEE**)

—— This page intentionally left blank ——

CHAPTER 5

GENERATION OF BUSINESS MODELS FROM PGR-BASED REQUIREMENTS

"What I need is perspective. The illusion of depth, created by a frame, the arrangement of shapes on a flat surface. Perspective is necessary. Otherwise there are only two dimensions. Otherwise you live with your face squashed up against a wall, everything a huge foreground, of details, close-ups, hairs, the weave of the bedsheet, the molecules of the face. Your own skin like a map, a diagram of futility, crisscrossed with tiny roads that lead nowhere. Otherwise you live in the moment. Which is not where I want to be."

– Margaret Atwood, **The Handmaid's Tale** (1985)

5.1 INTRODUCTION

Business models (BM) play an ever more pivotal role in the development and continued management of information systems and technologies in organizations, especially for building bridges and enabling dialogue between business and technological worlds. Nonetheless, recent literature reviews on BM results show that there is no agreement on what a business model is, although some emerging common themes already exist [Zott, Amit, & Massa, 2011].

Furthermore, the business model artifact, as a conceptual tool that contains a set of elements and their relationships, expressing the business logic of a specific organization and the value it offers, is seen as crucial for improving the dialogue between business and information systems. Though it has had substantial attention from both academics and practitioners, its growing associated literature is still recent and dispersed.

Complementarily, when eliciting requirements for the design of information systems, business goals and rules associate with business process use-cases to compose a business model base structure, driving and supporting an organization's strategy. The knowledge represented in terms of goals, rules and processes can make reengineering tasks more systematic and effective [Yu & Mylopoulos, 1994].

Whether it involves the development of a new system or the reengineering of business processes, decisions about what goals to pursue and on selecting the appropriate strategies to achieve them are always vital. The discovery of goals and rules is part of requirements elicitation, recognized as one of the most critical activities of information systems development, with many prescribed methods and techniques. However, existing methods for relating business processes, goals and rules (PGR) are scarce, dissonant, poorly grounded or highly analyst-dependent.

Our work in generating a business model in ill-defined contexts, within a RUP-based approach and grounded on reference model representations, stands as a contribution inside this topic (Section 4.3). The use of process-level use-cases, together with business goals and rules associated information (PGR), allows developing an activity direct-mapped business model to present to stakeholders for validation. The use and adaptation of 'standard' methods and techniques to infer goals and rules requirements from scenarios and process-like diagrams, mapping backwardly the traditional business to process workflow, constitutes our hypothetical approach for a better and continuous alignment between business and information systems, with improved traceability.

Alongside, there are challenges on how to tackle the business semantics, which are often culture dependent, and the intercultural competencies necessary for requirements engineers operating in these environments. New techniques need to focus not only on the target technical system but also on the interplay between business and system functionality.

In this context, the four perspectives of the balanced scorecards present themselves as an improvement on purely linguistic-based approaches, helping to bridge the communication gap between client stakeholders and business analysts, while improving the understanding of interrelations between strategic goals and system functionality [Loucopoulos & Garfield, 2009]. Additionally, as balanced scorecards are the reference in strategy management, a combination of these three dimensions (processes, goals and rules, and scorecards) constitutes our hypothetical approach for leading to a stronger, more strategy-oriented, business model. While scorecards and processes represent, respectively, the business and functional dimensions, goals and rules can not be handled in a separate way, as together they represent the nonfunctional side associated to this concept of business model, and so are treated as one sole dimension.

Our approach tries to obviate to this, proposing a method to guide the business-process analyst in the elicitation of business goals and rules from process-level use-cases, and transforming them, in order to arrive at a business model representation. This later can then be presented to the involved stakeholders for review, validation and further negotiation. As the entire method follows a model-based approach, the changes agreed upon can be traced back to the original use-cases, allowing for requirements traceability and an aligned solution between business and information systems.

After an initial method design proposal and corresponding demonstration, we present an evolution for an integrated proposal, consisting in a representation metamodel and a tailorable method, to handle a three-dimensional, cube-like version, of a business model. Added to each of these, there are accompanying critical analysis and the development of a prototype support tool for the cube manipulation. Finally some conclusions are stated to complete this chapter.

5.2 A METHOD FOR THE GENERATION OF BUSINESS MODEL CANVAS

In the previous chapter we propose the adaptation of standard techniques to infer goals and rules from scenarios and process-like diagrams, mapping backwardly the traditional business to process workflow, which helps in building a business motivation model and defining a strategy for an information system. With an approach based on a Business Motivation Model (BMM) representation and guided by a Rational Unified Process (RUP) backward transformation-based from process to business, it could allow for better and continuous alignment between business and information systems, with improved traceability.

Following this research work, supported in the previously proposed PGR metamodel, we now aim to propose an innovative method to guide the analyst in the elicitation of goals and rules from use-cases (UC), and transforming them, in so generating a business oriented business model for an information system. This is achieved by combining the use of business goals and rules elicited from business process UC in a balanced scorecard (BSC) structure, and then performing their mapping to a business model canvas (BMC) panel. This method is tailorable within a Software Process Engineering Metamodel (SPEM) approach, making it adaptable to any project or team.

After outlining the design of the proposed method we present its demonstration within a partnership academic-industrial project, followed by a critical analysis on its design issues, strengths and weaknesses detected along its use in a real live setting.

5.2.1 Design of the Method

Our proposed method, here presented in a SPEM perspective (Fig. 5.1), is composed by two activities ("Inferring Goals and Rules from UC" and "BSC to BMC mapping") and involves three work products ("Top-level UC", "Balanced Scorecard", "Business Model Canvas"). The activities are sequentially performed in a way that an activity starts only when its predecessor activity has finished (as indicated by the «predecessor» dependencies), while using and producing (as indicated by «input» and «output» associations) artifacts.

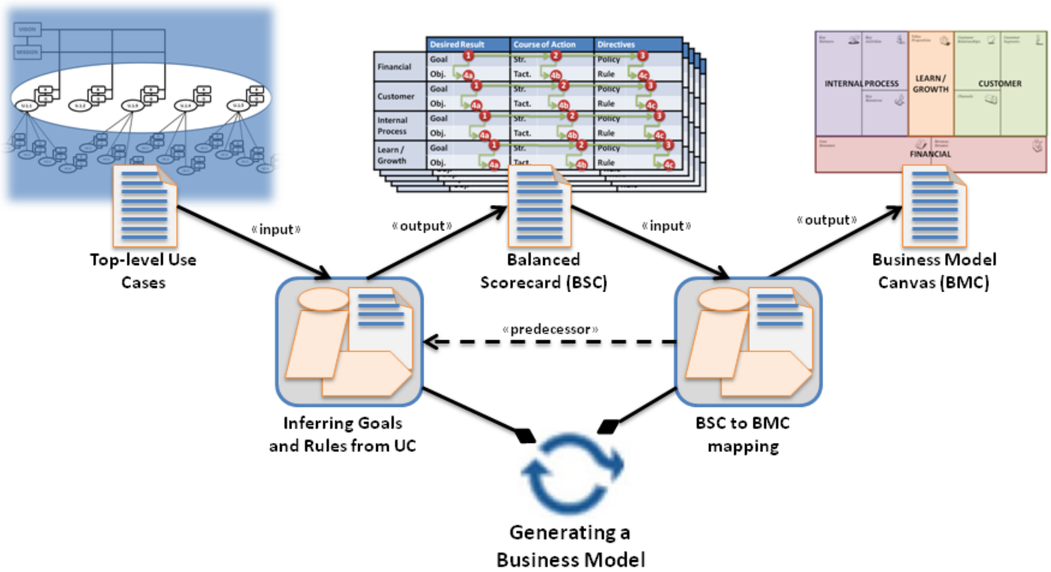


Fig. 5.1: SPEM diagram of the process for elicitation and generation of a BMC

The first activity aims to elicit and represent the PGR business-side information by following a 'standard' referential, spanning the four perspectives of the BSC, in so improving the consistency of the UC coverage. The second activity analyses and maps each previous elicited item in an adequate section of the BMC panel, linking them to the more abstract level of business modeling, thus delivering an integrated BMC to present to stakeholders.

Step I – Inferring Goals and Rules from UC

The first activity receives a set of UC and outputs a BSC, being composed by two sets of three tasks each. It starts with the elicited top-level UC for the proposed information system and involves two iterations, one for each UC and another one for each BSC perspective (Financial, Customer, Internal Business Process and Learn & Growth) with the added BMM representational elements (Goal, Objective, Strategy, Tactic, Business Policy and Business Rule).

Inside the double-iteration, there are the first three tasks to be performed (Fig. 5.2 and Fig 5.3) covering the elicitation of goals and rules, with its associated strategies and policies, for each root UC:

1. Envision the UC associated Goal;
2. Determine its governing Strategy;
3. Associate the controlling Business Policy.

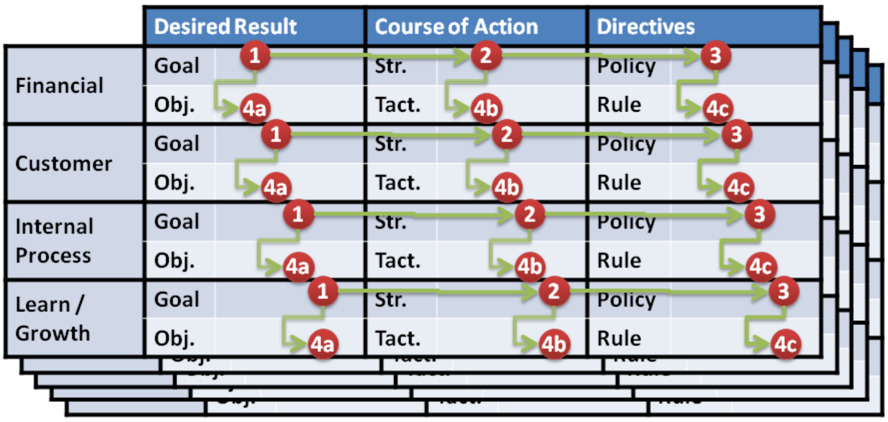


Fig. 5.2: Tasks for Step I of our proposed method

These are associated to the high-level, more abstract, BMM items, the ones preferable to start with due to their business-oriented nature, as they should be easier to elicit using the available business documentation. Also, depending on the project, the elicitation of these first elements could be enough for the generation of a high-level, more abstract, business model to present to stakeholders for review.

Thereafter, in other situations, a second three tasks sequence (Fig. 5.2 and Fig. 5.3), related to the more concrete and specific goals and rules items, can also be performed:

4. a) Define a (SMART[41]) Objective, associated to Goal;

4. b) Determine a Tactic, associated to Strategy;

4. c) Delineate a Business Rule item, associated to Business Policy.

[41] SMART: specific, measurable, attainable, relevant and time-based

Here, more detailed, concrete, information is needed from the project documentation to be able to elicit these items. Although not all fields are mandatory inside this BSC/PGR grid, it is important to fill the most part of them for achieving higher system specification coverage, especially to support future implementation purposes.

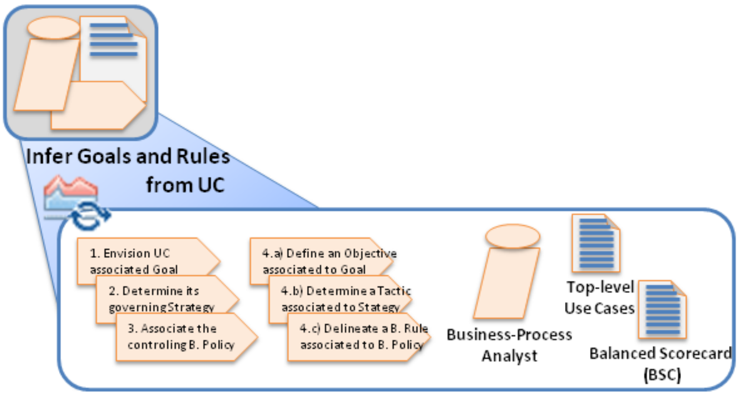


Fig. 5.3: SPEM Activity "Infer Goals and Rules from UC" constituents

All these tasks follow on the guidelines and checklists from RUP and BMM, as referred in the previous chapters 3 and 4, for the elicitation of business goals and rules, and its associated inner constituent elements. Any further knowledge of other associated techniques by the analysts involved, as well as heuristics associated to their previous experience in the specific domain of the project, are valuable to aid in these tasks. All referred tasks are proposed to be performed by a business-process analyst role.

Step II – BSC to BMC mapping

As stated earlier, BMC stands as one of the preferred tools for the generation of business models, especially in business related audiences. Also, BMC relates its roots with BSC, an also popular strategy performance management tool.

According to the Business Model Ontology (BMO) work [Osterwalder, 2004], the nine elements of the BMC relate directly to the four perspectives of BSC (Fig. 5.4), namely:

- Financial – Cost Structure and Revenue Streams;
- Customer – Customer Relationships, Channels and Customer Segments;
- Internal Business Process – Key Partners, Activities and Resources;
- Learn & Growth – Value Propositions.

Therefore, our proposal for the mapping of the sentences from our BSC-like structure to the BMC panel follows on this same line of thought: each sentence in BSC maps to one or more correspondent element in BMC. When there is a correspondence to two or three elements in BMC, any necessary decisions to choose on which specific element the sentence maps or on the separation in two or three statements, is responsibility of the analysts involved, whether based on the BMO guidelines or on their own business knowledge and heuristics.

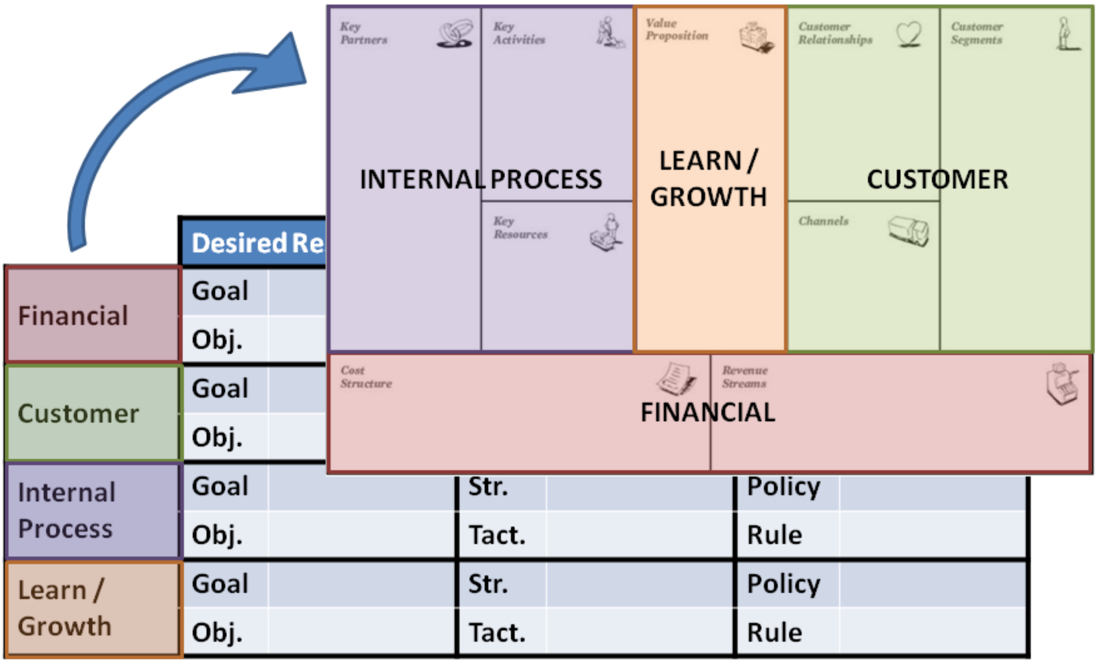


Fig. 5.4: BSC to BMC mapping in Step II

So, the second activity receives the previous set of BSC and outputs a BMC, being composed of four tasks (Fig. 5.5). All referred tasks are also proposed to be performed by a business-process analyst role.

Surely there can be some overlaps between some elements, for example, the value proposition is closely related to any other element, especially the customer ones, so it is not impossible for mappings to occur outside elements other than the ones proposed in literature. Other existing solutions in the business market propose slightly different mappings, but overall it depends on each particular case business type, and the informed or heuristic-based decisions of the analysts. All these matters should be the target for a final round of negotiations with the stakeholders, having the possibility for backtracking any changes made back onto the use-cases.

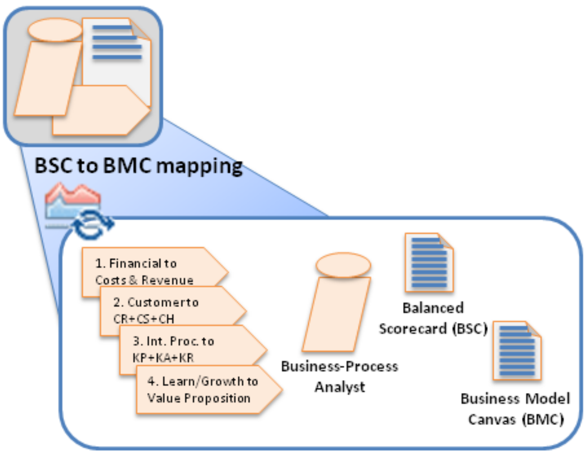


Fig. 5.5: SPEM Activity "BSC to BMC mapping" constituents

Several iterations of this entire process should be performed until all parts are comfortable with the obtained solution. The generation of a business model through the use of this first PGR step serves two purposes: on one hand it allows to communicate with the stakeholders of the project in a more business-like language, in a format that is familiar to them; on the other hand, it allows for a direct alignment and enabled traceability between the elicited use-cases for the proposed information system, and the business model to be analyzed and validated by the stakeholders.

5.2.2 Demonstration Case

The demonstration project is a new job matching and e-learning, cloud based platform, sponsored by technology-leading European companies, which aims to recognize and develop talents on the skills searched by employers, in order to tackle the shortage of professionals in technical areas. The core ambition is to offer targeted online education programs to improve ICT-skills, leveraging demand and supply on the European ICT job market, for science, technology, engineering and mathematics (STEM) people, preparing graduates for an industrial career and offering new skills and capabilities to empower current workforce.

The proposal is to create a business model that promotes creating a social network for e-learning-based talent matching, operating in Front-Office model, where each national office is responsible for its regional activities. UML use-case diagrams, as a powerful and useful technique for capturing the system's requirements at the first high-level of abstraction, were used to elicit the functionalities and the entities that maintain and interact with the platform services. Regarding the project complexity and for the sake of our technique, just the five top-level UC are considered to contribute to the generation of the high-level BM. After identifying the top UC (Fig. 5.6), the next step was to describe their behavior.

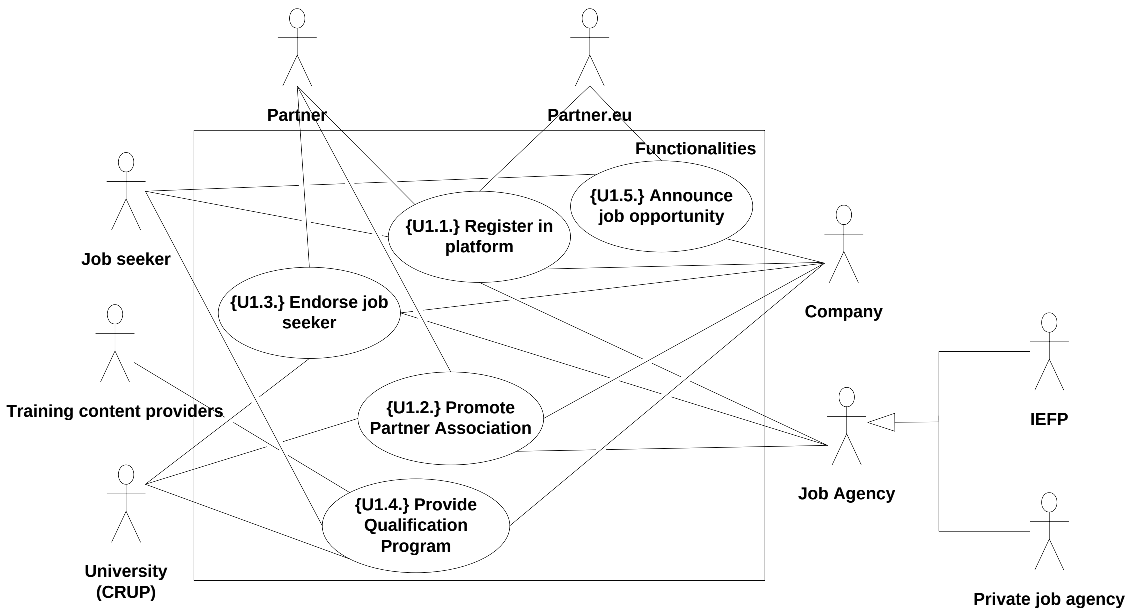


Fig. 5.6: Use-Case Model regarding the Project Functionalities

Although diverse forms of information are available in the project (informal text, activity diagrams, etc.), a structured description per UC during elicitation is not enough to generate the inputs for the BM as stakeholders needed. This due, it was decided to apply our developing technique in this live setting.

As described in the previous section, our proposal involves two activities which envision the filling of up to twenty four (four BSC times six BMM) statements, not all mandatory, per UC, followed by a mapping to a BMC panel. As previously stated for the decision of filling each statement, to be mandatory or not, rests in the hands of the analysts' team, as this method is composed of a flexible set of steps and techniques that can be tailored to each project and according to each situation. It may be given emphasis to the high order statements or be decided to go deeper in the goals and rules derivation, depending on the intended depth and detail to obtain, and also on the complexity of techniques to use, all influenced by the time and resources available, while taking into account the experience of the analysts involved. At this point, the previously SPEM method content definitions (work product, role, and task) for each activity are tailored and applied in this project by the development team.

Activity for Step I – Inferring Goals and Rules from Use-Cases

Although some projects require only the generation of the higher-level and abstract BM information, covering only the filling of the three more abstract BMM items, in this project the analysts involved opted to tailor the first activity of the previously defined process (Fig. 5.1) with the total three plus three tasks for the first activity (Fig. 5.3), performing the complete step with all the six tasks. As an example we present all the abstract and concrete items filled with business information for the UC '{U1.1} – Register in platform', regarding the BSC Financial perspective (Fig. 5.7):

1. In UC {U1.1} the goal to the Financial perspective includes the registration of the most possible amount job seekers on the platform;

2. The governing strategy is in the partnership between universities and employment agencies to reach a wider audience to register in platform;

3. Business policy indicates that all STEM ex-students with some specific requirements should be contacted for interest in registering in the platform;

4.a) To achieve the associated goal, the objective is to register at least 1.000 job seekers annually;

4.b) To create the partnership to allow more registers the tactic is to be present and publicize at job fairs, universities meetings, etc.;

4.c) The associated business rule to the policy for contacting ex-students is that only one mail should be send to the ex-students/graduates to prevent spam practices.

		Desired Result		Course of Action		Directives	
{U1} Register in Platform	Financial	Goal	Register the most possible amount job seekers on the platform	Str.	Partner with universities and employment agencies to reach a wider audience	Pol.	All STEM ex-students and <6 months before graduation students should be contacted for interest in registering
		Obj.	Register at least new 1000 job seekers annualy	Tact.	Organize mailing lists to contact eventual job seekers Be present and publicize at job fairs, university meetings, etc.	Rule	Only one email should be sent to each student/graduate, so preventing spam practices Only STEM-origin, European-able workers should be registered.

Fig. 5.7: Financial Perspective with BMM items for {U1} Register in Platform

After populating the entire BSC grid for this first UC, in the remaining three perspectives, this first step iterated through the other four UC, populating all the BMM/BSC items accordingly. Following, the second step was to map these sentences to the BMC elements.

Due to the strong business-orientation of the first three tasks, it was decided to iterate individually each one at a time, eliciting first all the goals for each top-level UC, then passing on to all the strategies and then to the business policies. This allowed keeping the mindset about the business guidelines for each BMM item throughout all the UC analysis. Next, the remaining three tasks were grouped and iterated sequentially, as their composition is somewhat easier to develop, highly-dependable of the previous elicited items (Fig. 5.8).

This activity is itself iterated through the four dimensions of the BSC grid, which after being populated with all the elicited BMM items for all the UC, five in this case, was ready to follow as an input to the next activity.

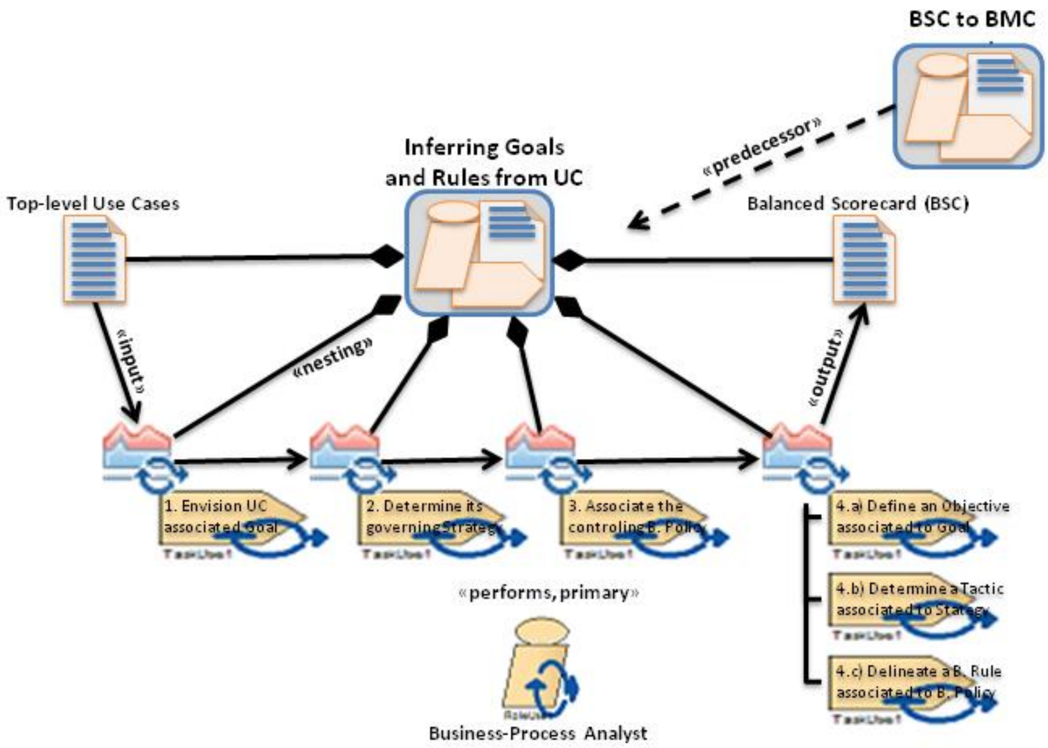


Fig. 5.8: Tailored development process for activity "Inferring Goals and Rules from UC"

Activity for Step II – BSC to BMC mapping

Although all six BMM items were elicited in the first step of this project, in this first round of the method execution, only the three more abstract items of each BSC perspective were considered relevant to be transposed to the BMC elements. This was due to the high-level positioning of the stakeholders to whom the BMC was addressed, and also to the shortage of information and ill-definitions of the project.

Although this second activity has less flexibility for being tailored, each task can be enriched with guidance information for its execution depending on the project's characteristics. In this case the four available tasks (Fig. 5.5) were sequentially iterated through the BSC dimensions (Fig. 5.9), splitting and adapting items as needed to map in the BMC sections. This activity is itself iterated through all the BSC cards served as input from the previous activity, giving origin to a complete BMC panel.

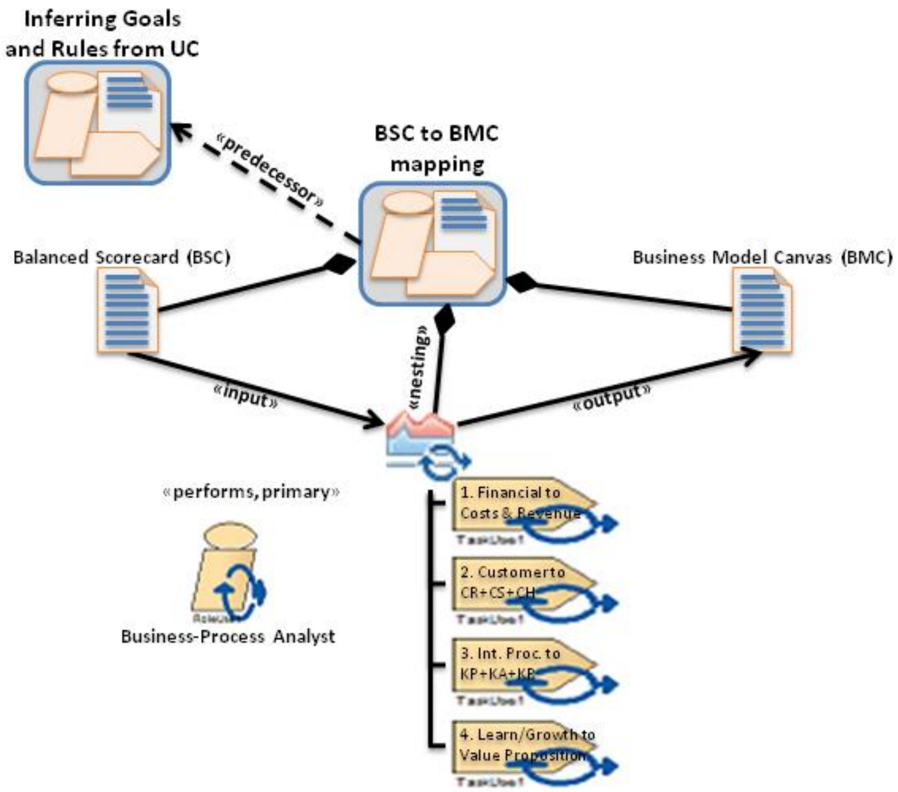


Fig. 5.9: Tailored development process for activity "BSC to BMC mapping"

Again as an example, regarding 'UC {U1.1} – Register in platform', its Financial perspective was categorized in the Cost Structure, the Customer in Channels, the Internal Process in Key Activities and the Learn/Growth in Value Propositions, all respectively. A discussion on sentence splitting to cover more specific aspects of dual elements, for example revenue-side information of cost-oriented UC, modification of the original sentences or their repositioning in different elements, is scheduled at a later stage within the stakeholders negotiation.

The solution business model for this project, after the first iteration, generated by aligning the resulting trios (PGR) with the BMC (Fig. 5.10), was presented to the project stakeholders and used as a token of discussion in the following meetings adjourned.

5.2.3 Discussion

This proposal for generating a business model through the elicitation of business goals and rules from process-level use-cases has a dual standpoint. While it relies on 'traditional', established reference techniques and model representations, it also innovates on the organization and relationship of these to achieve our hypothetical solution. First of all, it realizes the second research goal of this thesis, by providing a guiding method for the generation of business models based on business requirements. Also, due to the lack of consensus in the definition and representation for the concept of a business model, it contributes to the continued use and consolidation of a set of associated reference models within the research community. Last but not least, due to its model-based approach, it can guarantee recorded formal and quantified evidences, that all elicited elements are traceable and with a measurable impact.

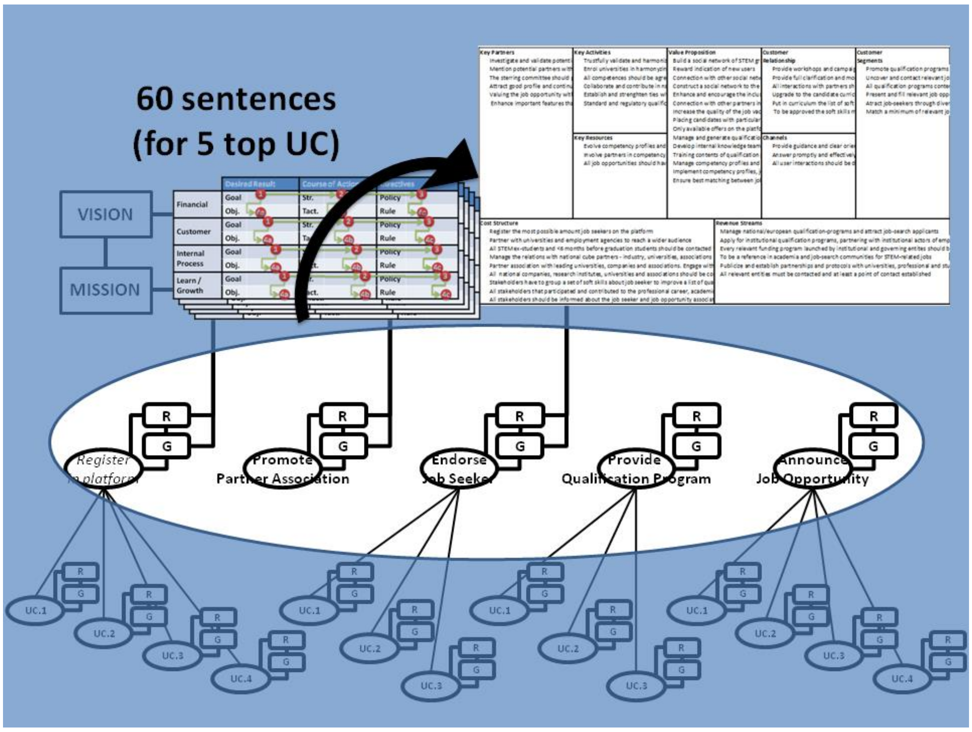


Fig. 5.10: Overview of the process for the current project

The basis on established references reinforces the solution and the original organization allows for advances in research. Flexible and open solutions maintain an open door for dialogue between business and information systems, leaving to analysts the choice for using other complementary techniques at some points, as also some room for negotiation with stakeholders. While many business related solutions are heavily dependent on the analysts' experience and flair, which makes them difficult to be quantified or strictly proven, one can say that this current approach takes a step forward by adding some formality to this process. Although, in this type of setting, one can never formally guarantee to have achieved better results with the use of one or another method, we can at least state that the same results plus an added reference support are accomplished, which constitutes a step forward in what regards the existing solutions.

The proposed method supports the effectiveness of BSC due to the individual connection to each UC, and the associated elicitation of goals and rules elements, as they are segmented in the four BSC perspectives. Moreover, it allows negotiating the positioning of this previous information in the BMC elements, and supports the business/IS dialogue by triggering interactions from the stakeholders.

One of the purposes for the proposal of this method was to overcome the lack of data in the initial documentation and more specific inputs from stakeholders to define the business requirements. The use of mixed techniques between business and IS allows analysts to, simultaneously, advance in the development of the information system and communicate with stakeholders, which helps in overcoming those issues. Its application represented a leap forward in this concrete project, as it sparked the dialogue between the different stakeholders, which were in an entangled situation due to the lack of elements for communicating.

The solution obtained already had a positive impact in the development, sustainability and evolution of the project. Nevertheless, the players in the project were aware that creating a BMC alone would not translate immediately into business success. The results have been promising so far, with positive feedback from involved stakeholders and research peers, which were due to the new approach and could not have been achieved solely by the use of the BMC, as initially attempted in the project. Although the adopted approach permitted to obtain a satisfactory solution in this specific case, the research associated with this thesis is still pertinent within a global and general approach taken in any other case.

On another point, according to the RUP referential, the types of tasks associated to these steps are to be performed by a person with a business-process analyst profile, which seems too broad. In this project we observed that the first step required a more IS-oriented profile while the second step required a more business-oriented profile.

The amount of human resources involved needs to be taken into account, as there is the excess work of eliciting twenty four times more information for each UC (four BSC perspectives times six BMM items), which increases the complexity and length of the project. Nevertheless, if the resources are available, a comprehensive approach on the goals-rules tree should be performed, ensuring a better validation of the root use-cases and goals-rules associated information.

The heavy reliance on the analysts' knowledge and their business heuristics calls for a stronger validation, with the need to conduct further studies on the associated tasks in each step. These should rely on diverse audiences, within real projects with business and information systems professionals, and in educational context with students in the IST area. Also, involved in our own research group internal work, this research aims to evolve the current steps to a complete PGR-based method, integrating with other internal methods and technologies, focusing on the PGR trio associated information.

The need for greater formality on the process representation, especially for the lower-level items, led us to use SPEM. Moreover, after an initial phase where the only tools used to aid in this process were some spreadsheets, we grounded the decision for the development of a prototype tool support on this SPEM specification, in order for this research to evolve. This prototype is presented in the last section before the Conclusion, at the end of this chapter, with examples of the use of this tool available in Appendix A.

5.3 THREE-DIMENSIONAL BALANCED SCORECARD BUSINESS MODELS

Our proposal on the use of process-level use-cases, together with business goals and rules associated information (PGR), integrating functional and nonfunctional perspectives, allows developing an activity direct-mapped business model to present to stakeholders for validation (previous section). Nevertheless, that proposal is not so easily understood by different users, due to the diverse concepts involved and representations of the different perspectives.

An increasing interest over the BSC when linking organizations strategy and operations [Atkinson, 2006], and recent research results in its contribution inside the information systems area in order to strengthen its alignment with business [Gyory *et al.*, 2012], caught our attention. It stimulated the need to explore and detail these issues, making clearer the importance and relevance of the BSC contribution to our proposals.

The relation between functional, nonfunctional and BSC perspectives can be viewed as a three-dimensional reality, lifting views from plain one-dimensional prescriptions [Abran & Buglione, 2003], while added to the comfort perceptions presented to users by cube-like structures ([Stroian, Bourque, & Abran, 2006], [E. G. Hansen, Grosse-Dunker, & Reichwald, 2009]). This leads us to propose a similar approach regarding the generation of a business model.

Recurring to the, requirements-based, two-dimensional PGR metamodel and by extending it with BSC as its third dimension, we propose a solution for the representation of the three related perspectives in a cube-like structure, following those three axis, the PGR-BSC Cube.

This can aid in the definition of general approaches and overall principles to assist analysts in better handling and understanding domain needs, as, in the end, only the experienced analyst perceives intuitively which method or technique is effective, in each circumstance, and applies it. Again, to support this solution, a specification of the method in SPEM is presented and then tailored to a specific visualization and processing objective, according to the needed information elicitation to fill-up the cube. This process allows for further deployments according to its intended use.

So, the main objective for this cube is that it can be understood and handled by different users, for diverse visualization and processing purposes. Also, as the entire method follows a model-based approach its operationalization can be traced back and forth through the different perspectives, allowing for requirements traceability and a business/IS aligned solution, while smoothing future tool development. Although this thesis focus is not about usability and graphical user interfaces (GUI), these topics raise important concerns as to the added value of using models, the advantages of working with a tailorable method and in delivering solutions which allow for its users to take full advantage of these properties. Moreover, due to the model-based approach followed on the design of these methods, tracing back any element within the model is naturally full-supported.

Following, we present the details of our PGR-BSC Cube proposal for a metamodel on its representation and associated cube structure, added to the specification of a SPEM tailorable method to handle their visualization and processing. Also, we discuss and frame our proposal inside its related work, by applying it twice in revisiting the previous project for the elicitation and generation of a BMC, where the method is tailored to specific visualization and processing objectives, according to each intended use. Finally we analyze some learnt lessons and the added contributions in this topic, while envisioning the future work ahead.

5.3.1 Design of the PGR-BSC Cube

Our proposed approach for a three-dimensional, requirements-based, balanced scorecard business model, from now on designated as the PGR-BSC Cube, is twofold. On one side, there is a metamodel for representation of the:

- processes (one face, in the form of use-cases);
- goals and rules (other face, with its constituent details);
- four dimensions of the BSC (third face of the cube).

Complementarily, we detail a method to handle the generated cube-structure, with its constituent activity, and associated tasks, work products and roles. This method is first specified, and then further tailored and organized, using the SPEM specification for soundness and clarity reasons.

PGR-BSC Cube Metamodel

The PGR-BSC Cube follows on our previous work in combining process use-cases, widely accepted on the functional side of requirements elicitation, with business goals and rules, as one prominent solution on the nonfunctional counterpart, in a PGR metamodel. These are then extended with the BSC information, relating them with the previous metamodel constituents (Fig. 5.11).

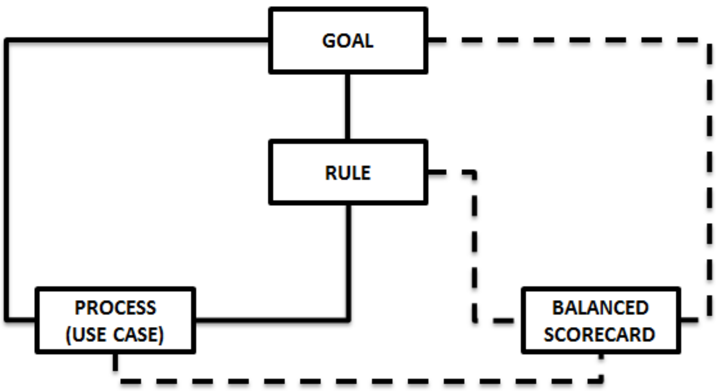


Fig. 5.11: PGR+BSC Simplified Metamodel

This metamodel allows for the representation of information relating to the three perspectives and, recurring to the associated cube structure, for its visualization according to the different intentions of its users (business-process analyst, systems analyst, software architect, etc.).

This relation was previously explored in the method for generating a business model through elicitation of business goals and rules from process-level use-cases, presented in the previous section. There, the BSC served as an intermediate stage to reach a business model canvas and the BSC elements were associated to each of the detailed individual elements of the PGR.

According to the PGR-BSC Cube simplified metamodel in Fig. 5.11, the BSC elements present themselves as a third standpoint connecting to the already existing two-dimensional realities of process and goal/rule. On the relation between these three perspectives an associated cube-like structure is built along three axis: use-cases, goals/rules and the balanced scorecard four dimensions (Fig. 5.12).

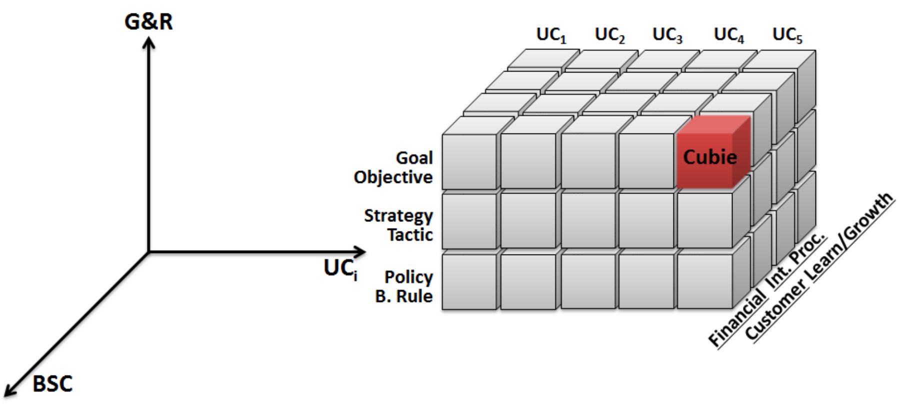


Fig. 5.12: (UCi; G&R; BSC) axis and associated cube-structure

The length of the process perspective, i.e. the size of its associated cube face, is variable and depends on the number of top-level UC of the elicited information system structure (five in this example: UC_1, UC_2, UC_3, UC_4 and UC_5).

Regarding the goals and rules perspective, its length follows its detail constituents inherited from the BMM representation (three or six, depending on the desired level of detail), namely:

- Goal/Objective;
- Strategy/Tactic;
- Policy/Business Rule.

For the BSC perspective, its length is related to its four dimensions:

- Financial;
- Customer;
- Internal Processes;
- Learn/Growth.

Although these two last perspectives follow a more fixed length status, additional BMM, BSC or other related dimensions can be added to the PGR-BSC Cube as needed. The cube form allows for a quick adaptation according to each project's needs or research trends. The elements in the intersection of the three perspectives are named cubies.

PGR-BSC Cube Method

In order to handle the information in this metamodel and associated cube-structure, we propose the design of an activity (using the SPEM specification), with its required detailed tasks, work products and roles. The activity is respectively comprised of four tasks, four work products and a user role of business-process analyst (Fig. 5.13).

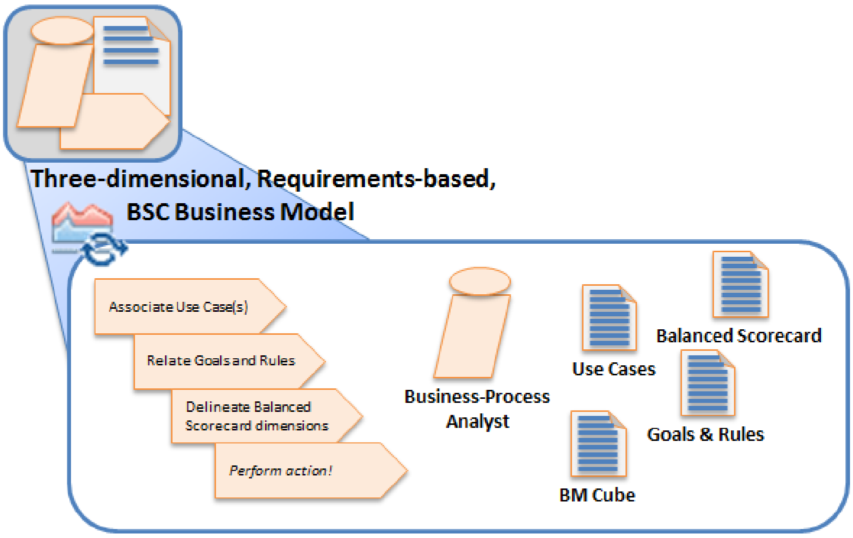


Fig. 5.13: Proposed activity with its tasks, work products and roles

Related to the work products, three are associated to each of the cube's sides and one is associated to the business model cube itself (the complete set of cubies). Regarding the tasks, three are associated to the handling of each cube's perspectives and one associated to the action to perform in each cube element (cubie).

So, the three first tasks involve the handling of the use-cases, the goals and rules, and the balanced scorecard, work products that form the structure of the business model cube, and the fourth involves the processing of the cubies. For the time being, the only user performing these tasks is the business-process analyst. The concrete tasks involved, respectively for each dimension are, respectively:

- Associate use-cases;
- Relate goals and rules;
- Delineate BSC dimensions;
- *Perform action!*

Due to this specification, the activity can be tailored in a SPEM process, operationalizing it according to the users' intentions. In this case we present an example (Fig. 5.14) of using the designed activity in a process to elicit the related business goals and rules from the initial associated use-cases, followed by the delineation of the respective BSC dimensions.

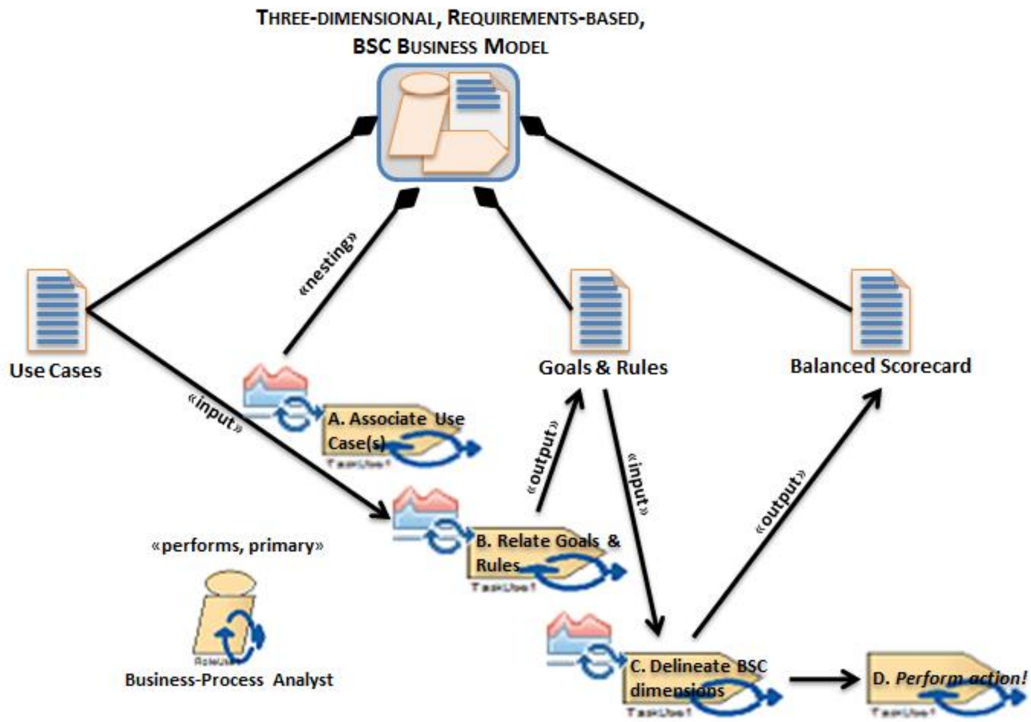


Fig. 5.14: Example tailored process for filling the cube

Each of the first three tasks in the activity can be organized and run in cascading cycles, handling the elements of each corresponding perspective, and at the end obtaining the necessary inputs for generating the intended outputs with the final task. The organization of the sequence of cascading tasks implies a corresponding visualization on the PGR-BSC Cube, as in Fig. 5.15, with the possibility to rotate each face accordingly. The capital letters (A, B, C) in the figure identify the order by which the tasks are organized in the tailored process.

That is, this type of cube structure and associated tailorable process adapt themselves to one another. When the cube rotates/switches, in any of its three axis, the sequence or cascading order of tasks to perform should be reciprocally switched too, in order to conform to the desired processing sequence or to the specific visualization perspective. The final task is the only one that should not switch, it is to be processed at the end, for each cubie. All-in-all there are six different possibilities for cube rotations, by performing different rearrangements of the three face-views (Fig. 5.15).

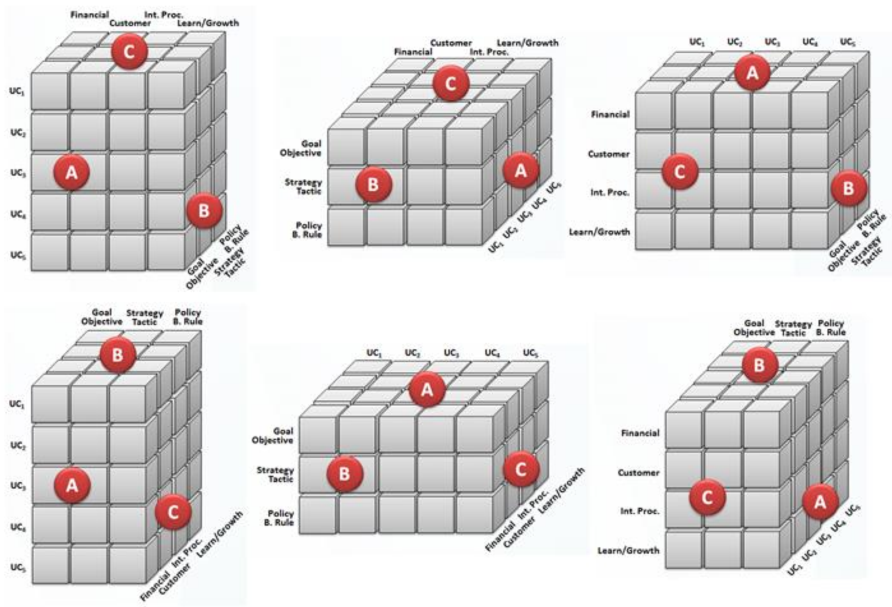


Fig. 5.15: PGR-BSC Cube perspectives (A: Use-Case; B:Goal&Rule; C: BSC)

This cube-structure and its associated method allow performing a transformation from three one-dimensional perspectives to one three-dimensional reality. Conversely, the opposite is also possible, as in the case of extracting a business model canvas (a one-dimensional reality) from the complete cube information for the three perspectives. So, the activity can be tailored in a manner to elicit the related business goals and rules from the initial associated use-cases, following the delineation of the respective BSC dimensions, or to run through the BSC dimensions, then the use-cases and extract the associated goals and rules statements.

5.3.2 Exploring the PGR-BSC Cube

Revisiting the AC.pt project (previous section) where a specification for the elicitation of business goals and rules from process-level use-cases, and their mapping to a business model representation was used, allows us to explore the possibilities of our three-dimensional, requirements-based, balanced scorecard business model proposal, the PGR-BSC Cube.

Following the SPEM process used in the referred project, depicted in Fig. 5.1, the generation of a canvas from an initial set of use-cases is done in two-step activities. First, the BSC were filled by inferring the business goals and rules from the process-level use-cases, and then, the canvas blocks were mapped from the BSC cells. Both activities have internal iterative cycles to handle the elements of the involved work products. This way, the BSC functions as an intermediary stage between the two steps, referring both to the PGR and the canvas elements, allowing for future traceability issues.

By using the PGR-BSC Cube structures and the associated tailorable method to perform these two activities, we can obtain the same or even better results, in a more user friendly way, through an improved, conceptualized, visualization of the entire method. Next we present the result of applying the PGR-BSC Cube approach to the two referred activities (Fig. 5.1):

- Inferring Goals and Rules from UC
- BSC to BMC mapping

From requirements to the PGR-BSC Cube

The first activity, 'Inferring Goals and Rules from UC', can be performed by our method through the use of one (Fig. 5.16) of the six available cube-structures (Fig. 5.15) and the corresponding tailored activity (Fig. 5.17).

The chosen activity guides the user through a cascading cycle in the following way:

- Select a use-case (A);
- Select a BSC dimension (C);
- Select a goal and rule representation (B);
- Elicit a business model statement (D).

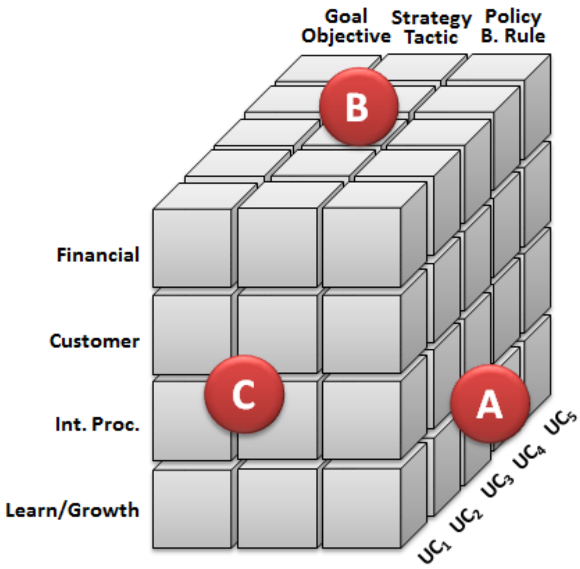


Fig. 5.16: Cube perspective (A-Use-Cases, C-BSC, B-Goals and Rules)

In this particular case, first the user selects a specific use-case then selects a BSC dimension, and finally a goal and rule representation. Having set the three perspectives, the user then performs the desired action, in this case eliciting a business model statement for the current cubie accordingly.

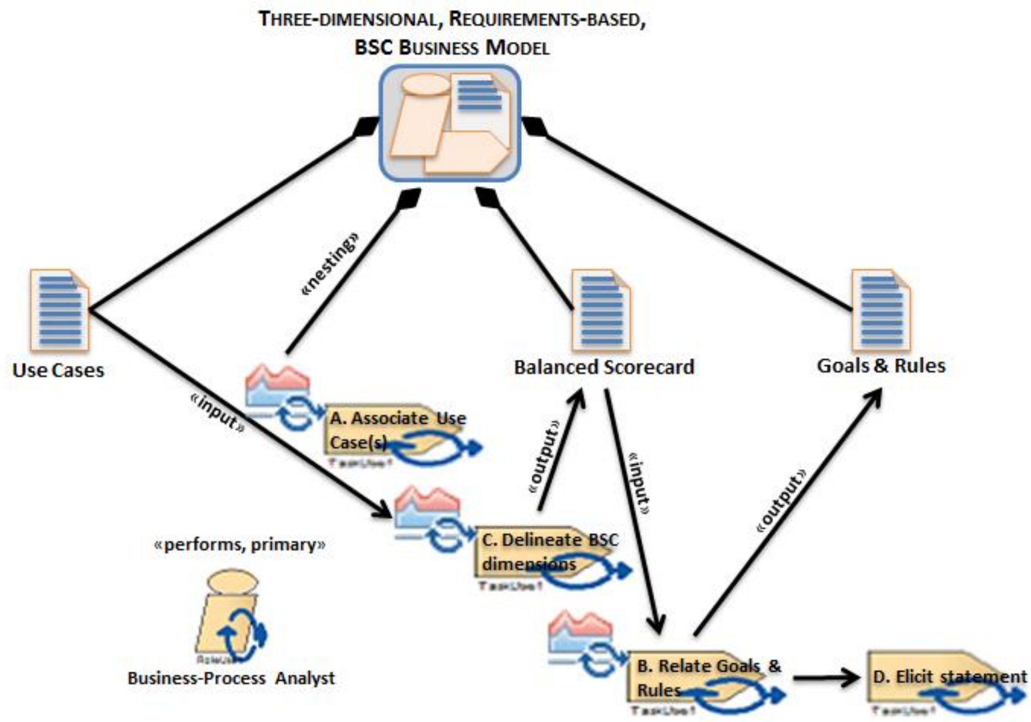


Fig. 5.17: Example tailored activity for eliciting business model statements

By iterating through these three perspectives in the stated order, the user will fill each cubie with an elicited business model statement, until all the entire PGR-BSC Cube is filled-up. The order of processing for the three perspectives was tailored according to our user's preference (A-C-B), but depending on the preferences of other users this could be easily switched, always remembering that the activity should be tailored according to the selected cube-structure.

The execution of the final task at each cubie is the responsibility of the analyst, while the other tasks can be more or less automated according to each method implementation. Also, although filling all the cubies is not mandatory, in this case it is important to fill the most part of then for a wider business model coverage. Moreover, in order for this task to be successful, the business-process analyst must possess sound experience and knowledge regarding guidelines and heuristics on the topic of goals and rules elicitation.

From the PGR-BSC Cube to the business model canvas

For the second activity, 'BSC to BMC mapping', our choice for one of the six available cube-structures (Fig. 5.15) can be seen in Fig. 5.18 and the corresponding tailored activity in Fig. 5.19.

Following the customized method, the chosen activity guides the user through a cascading cycle in the following way:

- Select a BSC dimension (C);
- Select a use-case (A);
- Select a goal and rule representation (B);
- Map a business model statement to the canvas (D).

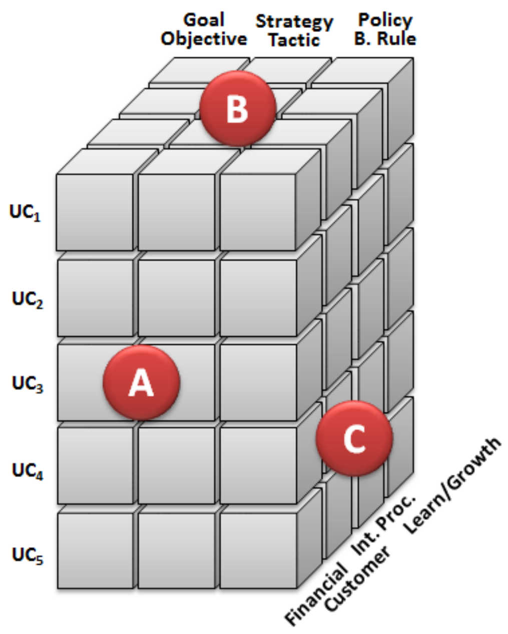


Fig. 5.18: Cube perspective (C-BSC, A-Use-Cases, B-Goals and Rules)

In this particular case, first the user selects a specific BSC dimension then selects a use-case, and finally a goal and rule representation. Having set the three perspectives, the user then maps the respective business model statement to the appropriate canvas block.

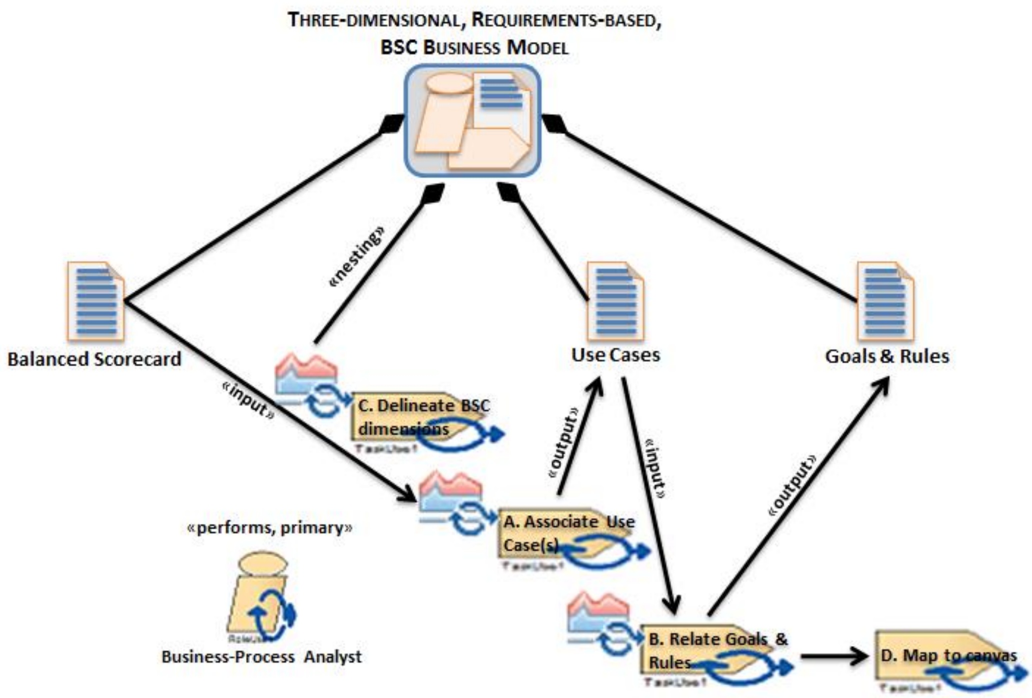


Fig. 5.19: Example tailored activity for mapping a business model canvas

By iterating through these three perspectives in the stated order, the user will map each cubie business model statement to a corresponding block in the destination canvas, until all statements are mapped and the canvas complete. Again, the order of processing for the three perspectives was tailored to our user's preference (C-A-B), but depending on the pretensions of other users this could be easily switched, as stated previously.

Once more, the execution of the final task at each cubie is the responsibility of the business-process analyst. In this case, mapping each element to at least a canvas block is mandatory, although sometimes the decision to which block to map or to divide the sentence in two blocks is not clear. Yet again, the analyst needs to use all experience and knowledge to make a correct decision.

5.3.3 Discussion

Cube structures are interesting in the way that they allow for better perceptions from the users and added flexibility in the techniques to work with them. This is important to aid in the work towards reducing the gap between experts and novices, by supporting practical roadmaps, frameworks, and guidelines that can be more linear and easily taught to students, novices or analysts outside a specific domain, as most approaches are hard to tackle at first and require a significant level of skill and expertise from the analyst to use effectively.

Related research

A major difficulty with organizational performance models in information systems, particularly in software engineering management (SEM), is to represent as many possible viewpoints quantitatively and in a consolidated manner. Meanwhile, keeping track of values for individual dimensions of performance in the selection of multidimensional models of performance found in SEM, instigated a review on visualization techniques [Stroian, Bourque, & Abran, 2006]. In fact, visualization means a graphical representation of data or concepts, which aids in having the right information at the right time, being crucial in making the right decisions. As many criteria can be used to evaluate the effectiveness of a visualization technique, a central one particularly fit to human perception is the use of three-dimensional cubes.

Another example of using three-dimensional perspectives is a generic model termed the "Sustainability Innovation Cube" which consolidated the research on sustainability and innovations into a coherent framework, clarifying existing relations in the field [E. G. Hansen *et al.*, 2009]. The proposed model did not quantify the sustainability effects but rather qualitatively depicted areas of sustainability potentials, while subsequently building on the current body of innovation methodologies. Such a generic conception facilitates its application to a wide field of businesses and products while at the same time enabling decision-makers to tailor the model by applying existing assessment methods.

An important part of enterprise business modeling is the creation of a high-level domain model that depicts the main business entities and their relationships that are of interest to an organization. The interrelation of the different perspectives, involved in the connection of business models and enterprise architectures, supports the alignment and tracing between their elements. This establishes the basis for a sound and effective business model, and allows identifying and specifying appropriate service-oriented architectural elements at enterprise level from business models [Jamshidi, Sharifi, & Mansour, 2008].

Also, as IT change processes affect an organization's enterprise architecture, it must also be mirrored by a change in the organization's business model, where its analysis may establish whether the architecture change has value for the business. One important reference in this specific topic proposed an approach to relate enterprise models specified in ArchiMate to business models, modeled using Osterwalder's Business Model Canvas [Iacob *et al.*, 2012]. As intended, the approach facilitates the tracing of business requirements captured by motivation and business models down to the design specifications, expressed as enterprise architecture models.

Contributions

The added value of the additional BSC dimensions in an existing functional and nonfunctional requirements metamodel specification, alongside its cube axis rotations and visualizations, as well as the SPEM activity tailoring, counts as an original step forward regarding existing solutions in this topic. This contributes in advancing the research knowledge, and supplies practitioners with a set of options for processing and visualizing a business model in linear ways.

The focus on the BSC allows for a traditional strategy management support, enhancing the interconnection between the business and information systems realities. Likewise, existing interconnection of BSC with both the problem-side elicited functional and nonfunctional requirements, as well as the management-side business model elements, permits widespread traceability throughout the entire solution and flexibility in applying any of the available transformation perspectives. The benefits of using the cube are reinforced by its continued use in projects within our research team, while the acceptance and publication of papers in relevant conferences sustain its theoretical arguments, namely within the associated research community.

Future work

Regarding future work, we aim to further detail each of these two specific scenarios proposed tasks, also presenting guiding heuristics and strategies for their operationalization. The exploration of the other scenarios and comparisons between the user's performance and its adaptability regarding the use of the different business model views is also in our future plans.

In another direction, the interrelation between business models and enterprise architectures is not so far off as it was in the past. As research from both sides matures connecting the two worlds becomes a real possibility. So, another future step will be to relate the PGR-BSC Cube with our logical architecture derivation method in a SOA environment (the PGR-SQC Cube) that is presented in the next chapter.

Moreover, the development of an aiding support tool to explore and process the different perspectives is essential at this point of this research. Due to the emergent availability of rapid-development platforms suited to the initial development of prototypes, standing as an interesting alternative for the initial evaluation and testing of our proposal, we decided for the current OutSystems proposal. Nevertheless, in the long run we maintain our intention to develop a support tool using the Eclipse framework, taking full-advantage of the metamodel and SPEM foundations of this work.

5.4 Tool Support: The PGR-BSC Cube Modeler

Again, according to the previously analyzed tools, our choice for the development of an agile prototype available through an aPaaS solution was the OutSystems Studio. After the initial development of a somehow simple solution, the PGR Modeler in Chapter 4, we now present the development efforts for a more complete and complex tool support, the PGR-BSC Cube Modeler. As we will see next, the number of entities, web screens and operations inside events for this PGR-BSC Cube is much higher than in the previous PGR Modeler.

Also according to our needs, we will be using mostly the Data and Interface layers, as even if the size and complexity of this particularly prototype is higher than the previous one, there is still no need for defining high-level business processes or broad business rules. Next we detail the relevant decisions made in these two layers for the PGR-BSC Cube Modeler prototype and additionally browse the working environment with some screenshots of the application.

5.4.1 Entities Structure

In order to implement an application to support the previously proposed metamodel, a set of corresponding entities were created, along with the necessary adaptations for the reality of a relational database. The list of entities is once more created in the Data layer of the OutSystems Studio, alongside the attributes of each entity and the automatic CRUD (Create, Read, Update, Delete) operations to perform with them (Fig. 5.20). Also included in this layer is the BSCDataModel entities diagram, presenting the relations between all entities (Fig. 5.21).

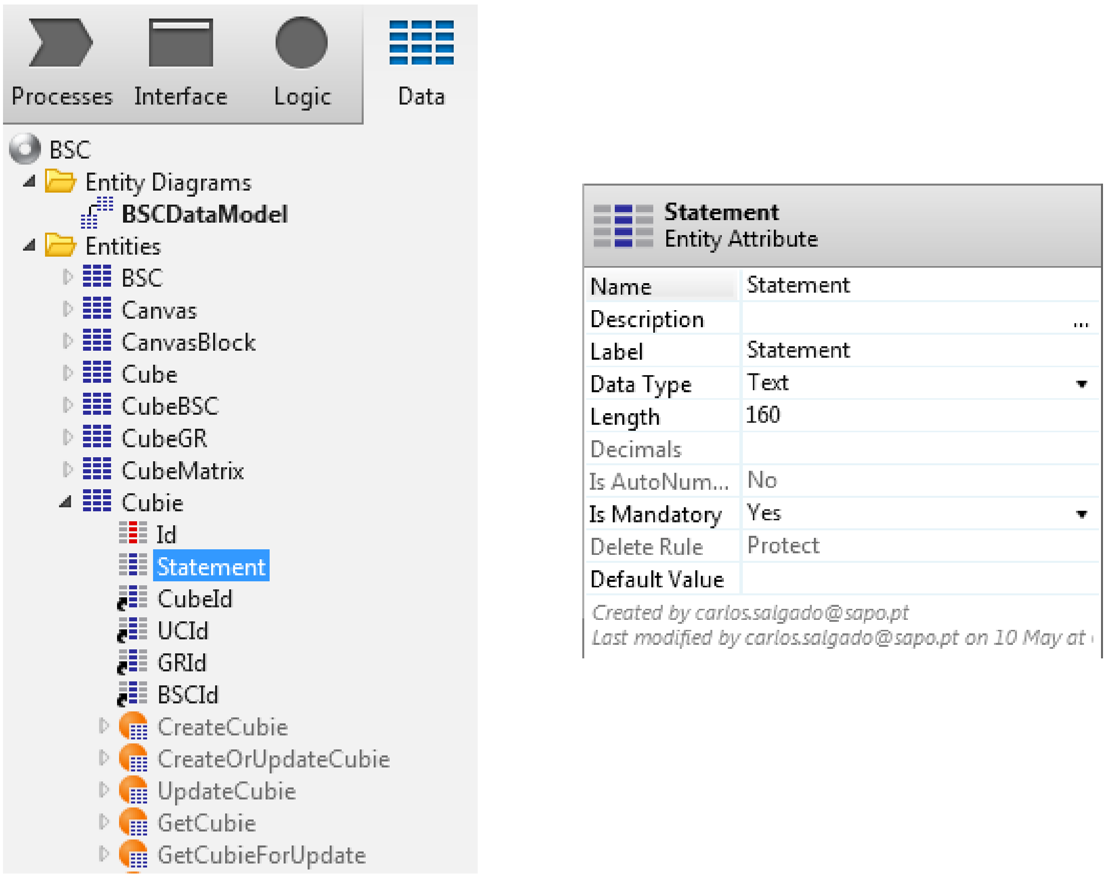


Fig. 5.20: Data Layer with the list of entities and detail for the Statement attribute of table Cubie

First of all, an entity Cube was created to allow for the creation and maintenance of diverse solutions for different projects, otherwise only one cube could ever be created with this application. According to the metamodel, the use-cases processes are represented in an entity UC, while the goals/rules and BSC dimensions have been disaggregated in two entities each. The entities BSC and GR allow for the base definition of any items related to these two dimensions, while the entities

CubeBSC and CubeGR associate to each cube only the dimension items to be used within its data. For example, we can define all the six BMM items in the entity GR, but only associate to a particular cube the top three, more abstract, items. As use-cases are directly related to a specific cube, they only need one particular entity.

Accordingly, the definition of every single element inside the cube is represented in the entity Cubie. Besides referencing all the previously referred entities (Cube, UC, GR and BSC), it includes a statement describing the associated goal/rule proposition, for a determined BSC dimension and a specific use-case. Moreover, in order to be able to map each cubie statement to element(s) in a business model canvas, two added entities are defined, the Canvas and the CanvasBlock. This last entity serves only for generalization purposes of the canvas blocks. As can be seen in Fig. 5.21, the Cube entity possesses external connections to the CubeBSC, CubeGR and UC entities (through the CubeId attribute), while the Cubie entity relates to those same entities but referencing their respective ids (BSCId, GRId and UCId), added to the descriptive attribute Statement.

The complete diagram for the PGR-BSC data model is presented in Fig. 5.21, where all the relations and dependencies between the different entities can be easily visualized. Contrasting with the PGR Modeler, the use-case entity cannot reference itself, as in this case we are only working with the top-level use-cases elicited for the information system. Note also that although the CubeID attribute in the Cubie entity is not structurally necessary (it could be obtained through the UC entity), its presence eases the development steps further ahead. The number of entities in the PGR-BSC Cube Modeler more than doubles the ones from the PGR Modeler.

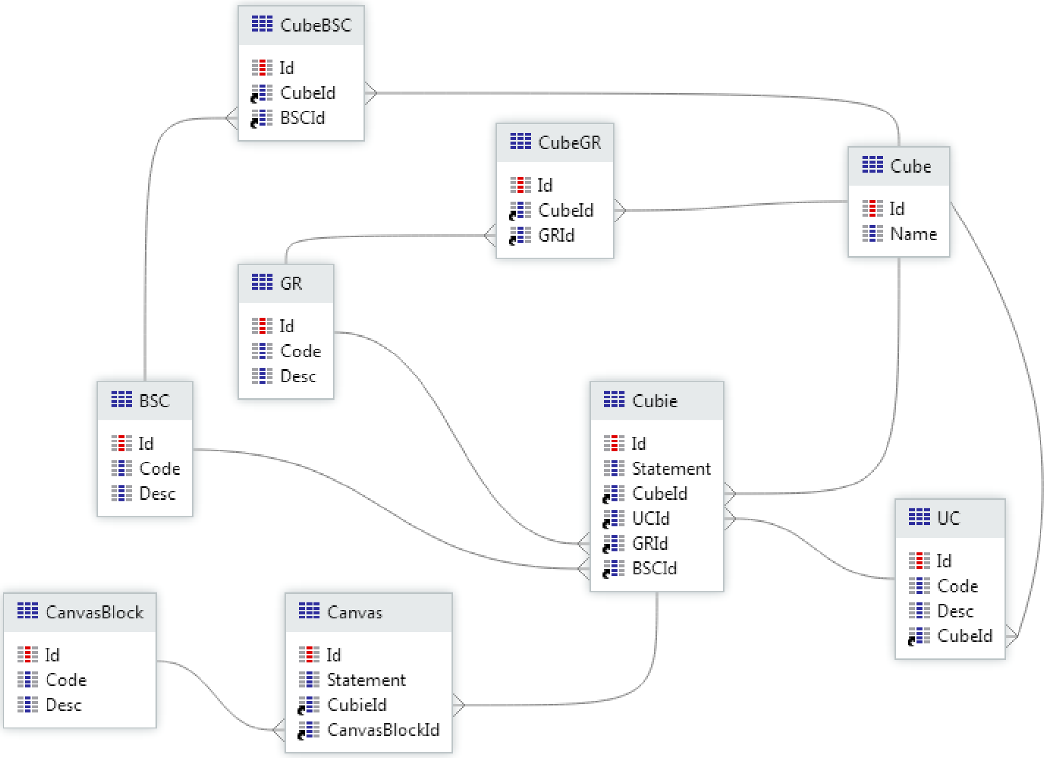


Fig. 5.21: Entities Diagram

5.4.2 Screen Flows

After finishing the entire entities structure, it is time to design the interface screens and the associated flows connecting them. Besides that, local entry variables can be added for control purposes and associated events created and/or configured for each screen. The most common of the events, already existent in every screen due to its importance, is the Preparation one, where specific business and/or technical logic is encapsulated to prepare data for the screen design. This is all possible through the Interface layer (left side of Fig. 5.22), whereas an example of a code 'flowchart' in the Preparation event for the CubeBSC screen is shown (right side of Fig. 5.22).

Again in the MainFlow, with an initial designed Main screen, the number of screens for this application also more than doubles the ones from the previous application. Once more, the use of the automatic scaffolding assisted in quickly creating screens to handle chosen entities, just by dragging and dropping each desired entity into the screen canvas space. This generated the typical listing screen for the elements of that entity, with associated filters, attribute sorting and others. Nevertheless, in this application much more manual configurations were needed afterwards.

For the base entities, by dragging a second turn the same entity in the canvas, a typical detail editing screen for the elements of that entity was also created automatically, with all the links between the two screens and all the needed validations. For the intermediate entities and the more complex relations, as between the Cube and Cubie, many adaptations were made afterwards on top of the pre-generated screens and on newly created screens that were needed (left side of Fig. 5.22). The complete set of generated screens is shown in Fig. 5.23, where the associated flow between the screens is represented by arrows.

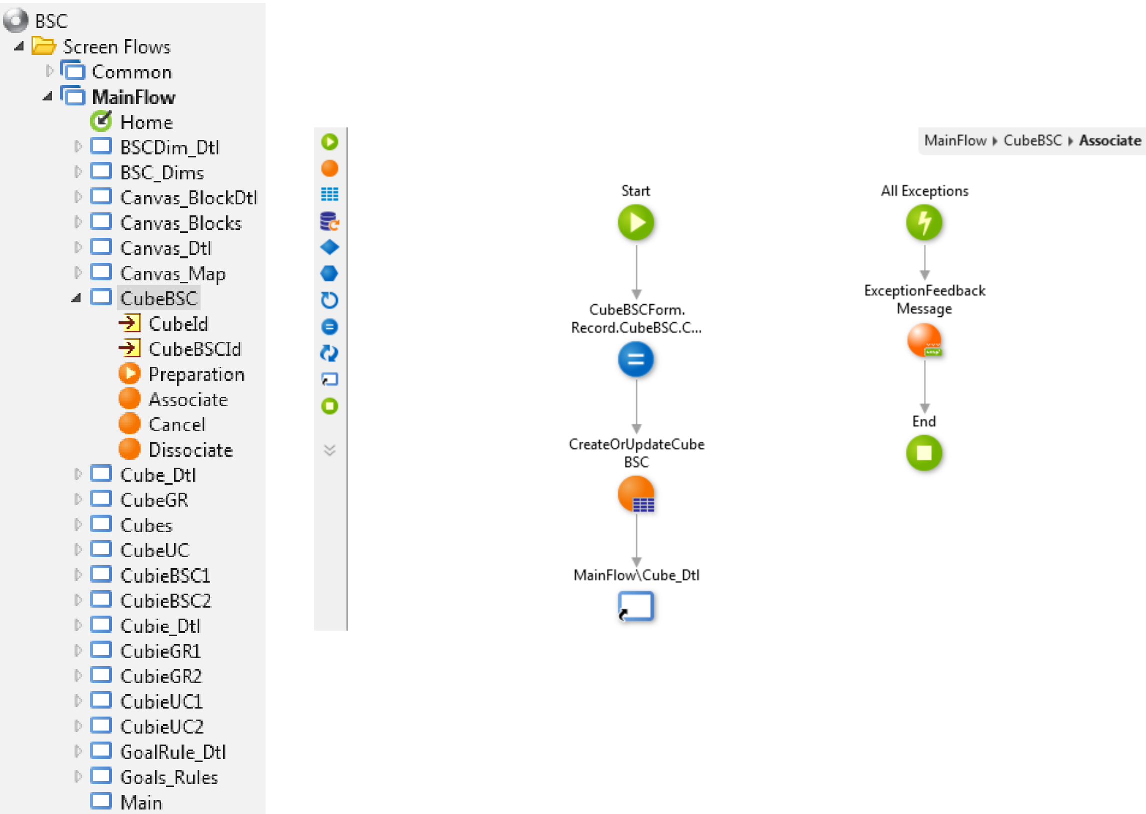


Fig. 5.22: Interface Web Screens, local variables, preparation and events (Associate detail)

From the Main web screen we can reach the base definition of the canvas blocks, items for goals/rules and BSC dimensions, and also visualize the list of existing cubes. From there and into inside a cube detail we can create a new cube, or associate BSC, goals/rules and use-cases (this last by newly creating each one), and access the list of existing canvas mappings in an already existing cube.

Going back, from the list of cubes we can view each cube in six different configurations, as referred in the previous section, depending on the orientation of the faces for the cube. Independently of the cube face, by clicking on a cubie element, we are able to create/edit a new statement for the chosen position and jump in to associate its mapping to any canvas element(s) (Fig. 5.23).

Regarding the diverse configurations made in the multiple events of this application, we highlight the one designed for the Preparation events for the six different web screens to simulate the cube faces rotation (Fig. 5.24). Besides the semi-automatic datasets, some If conditions and For Each cycles, and many variable attributions, with even a handmade SQL expression, were needed. The base configuration is the same for the six different screens, except some minor changes in the datasets entities, and associated adaptations in the attributions and in the SQL expression.

In what relates to the web screens, we highlight the Cube_Dtl, as it aggregates, besides the name of the cube, all the information related to the configuration of its three dimensions, involving the BSC, goals/rules and use-cases associations. Its design involves text labels, also recurring to variables, input boxes, buttons, dynamic tables and linked objects (Fig. 5.25). The existence of container elements allow for the correct positioning of several elements in one form.

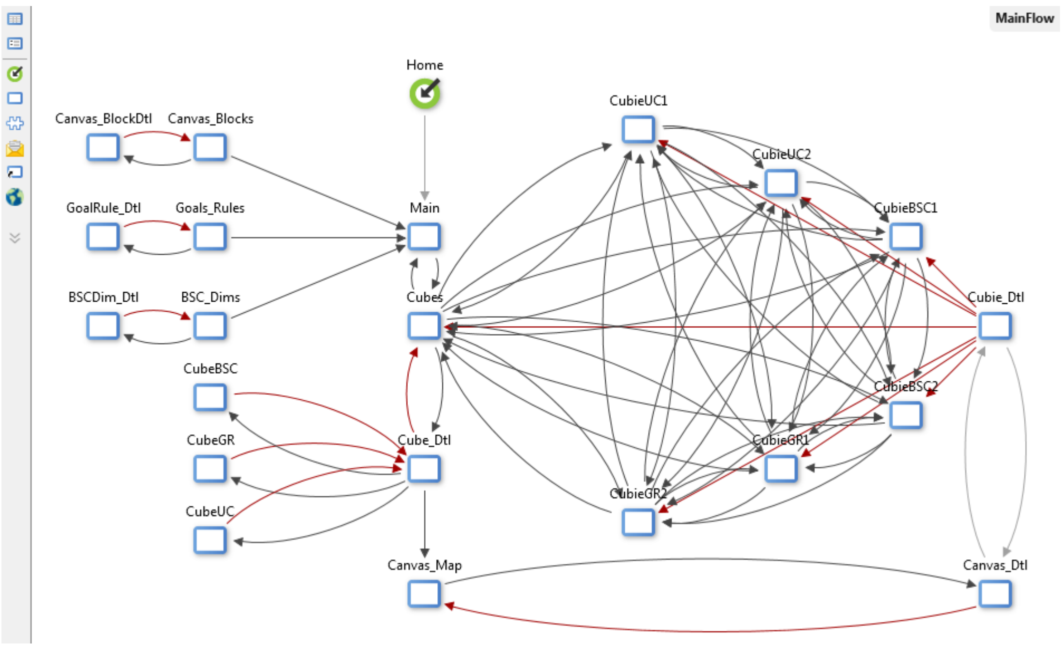


Fig. 5.23: Web Screens flow

Once again referring to the encapsulation of some needed business or technical logic in its events, besides the initial work set up in the Preparation event to execute prior to the screen visualization, when any of the buttons is pressed there will also be an associated portion of code to execute before continuing to the destination screen. Whenever a simple link is followed, it immediately hands out the control to that screen, where the Preparation event is promptly executed.

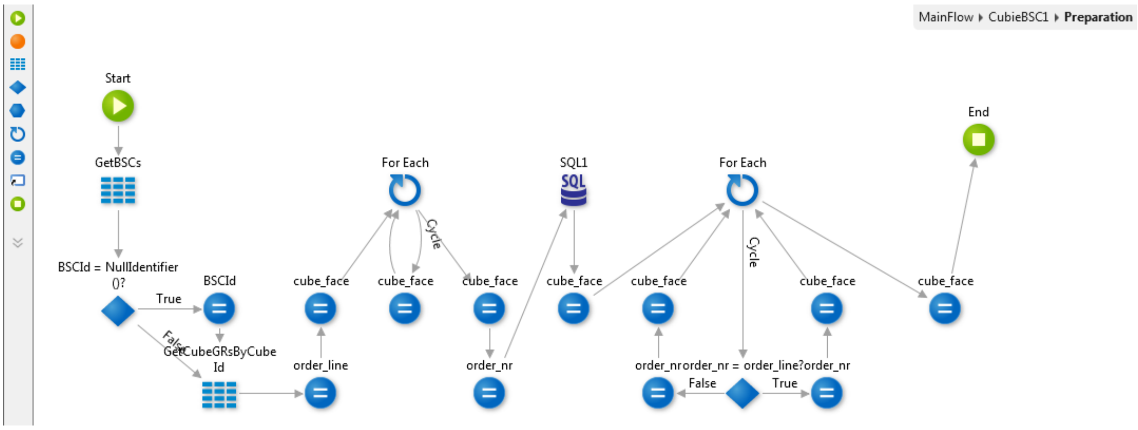


Fig. 5.24: Preparation logic for Web Screen

5.4.3 Form captions

Lastly, after the entities structured, and the screens designed and properly prepared, the application can be generated and deployed to the server. This concluded it is possible to immediately navigate the application and perform the desired operations. In the case of the PGR-BSC Cube Modeler application, the first operation should be to populate the items in the BSC and goals/rules dimensions, as well as the desired canvas blocks.

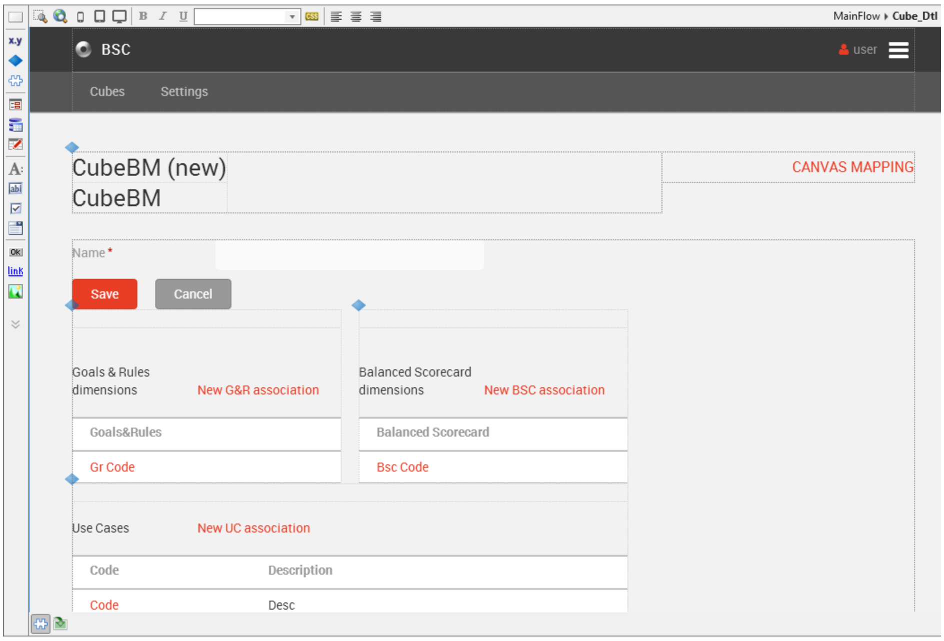


Fig. 5.25: Design mode for Web Screen

The next task is to create a new cube and associate the related items, creating the use-cases in the middle of this, to compose the different faces of the cube (Fig. 5.26). In the Cube BM form one can set the name of the cube and associate to it elements from three different dimensions: goals and rules, BSC and use-cases. From these three only use-cases are created in direct dependency from the cube, the elements from the other two have to be previously defined and then merely associated here. Lastly, the existing mappings from cubie elements to a canvas can be listed through the Canvas Mapping link.

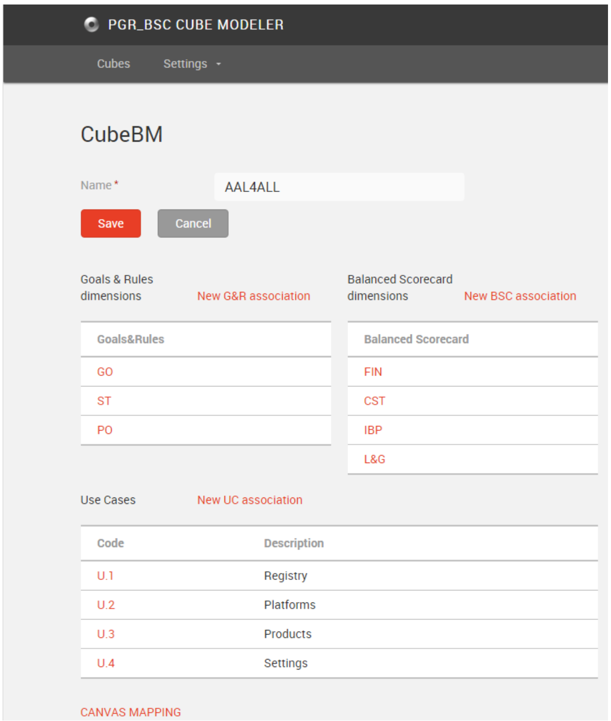


Fig. 5.26: Cube details

The main step of defining cubie statements can be performed thereafter, by accessing the cube through one of its six faces (Fig. 5.27) and repeatedly defining new elements, associated to a specific use-case, and determined goals/rules and BSC dimensions. Optionally, there is also the possibility to set the mapping of a cubie to any canvas element(s).

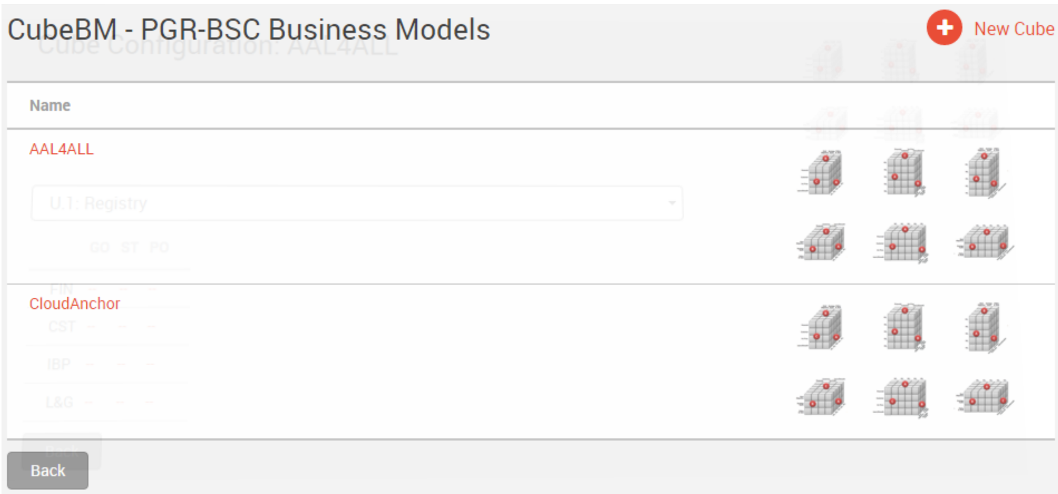


Fig. 5.27: Cube listing

So, besides the creation of new cubes, the Cube listing form allows accessing the previously referred details of a cube, by clicking on its name or on one of the six possible faces of the cube, by clicking on one of the six images on the top-right corner of each cube row (Fig. 5.27). The Cube face form (1 of 6) allows setting, editing or just viewing the cubie statements there available. The example of one of the six faces can be seen in Fig. 5.28, where one dimension is available through a combo-box that, when changed, refreshes the contents of the remaining two-dimension table presented below.

In this particular case the combo-box holds the use-cases items while the goal and rules, and BSC dimensions constitute, respectively, the table vertical and horizontal axis. Also, a simulation for the rotation of a cube can be performed by clicking in one of the six images on the top-right corner of the form, shifting the view throughout the six possible faces (top-right side of Fig. 5.28).

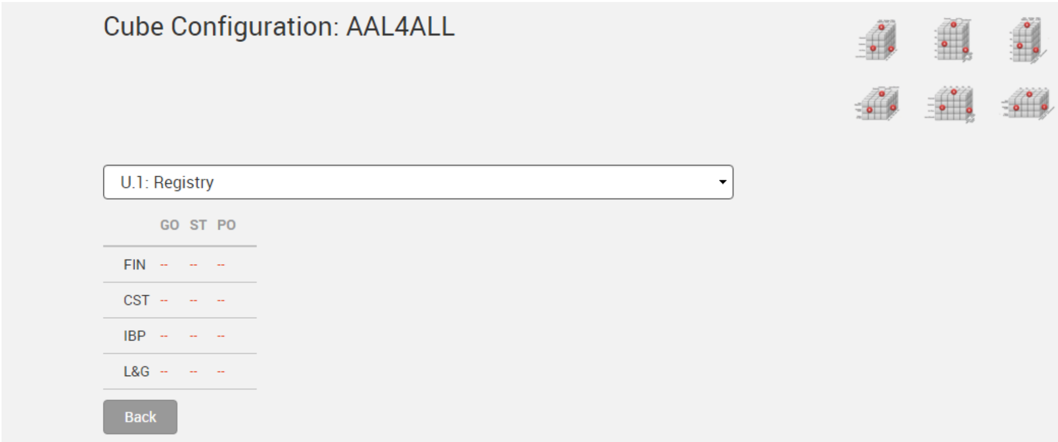


Fig. 5.28: Cube face (1 of 6)

The cubie statements are the core of the PGR-BSC Cube Modeler application, representing the equivalent to the PGR requirement statements of the previous PGR Modeler, but now with the added “flavor” of the BSC dimension. The Cubie detail form (Fig 5.29) allows setting a new cubie statement or editing an already existing one. It cannot be deleted, as the crossing of the three dimensions always exists, but it can be left blank if desired.

Fig. 5.29: Cubie detail

As referred in the previous section, a cubie statement can have one or more mappings to different canvas blocks, with each statement being adapted to the context of each block. This operation can be performed in this Cubie detail form, with the complete mapping accessible from the Cube detail form, also with an option for a complete canvas report/export.

Appendix A contains examples of the use of this tool regarding the project AC.pt, while the complete application can be accessed and freely tested in the following web address:

- https://carlossalgado.outsystemscloud.com/BSC/

5.5 CONCLUSION

The role of business models in information systems management has been shifting to embrace evermore the technological side, helping to shorten the distance between business and technology. Even so, there is no commonly built ground within its stakeholder's communities with much fuzziness still present in both academia and practice.

A relation has been established between business processes, goals and rules elicitation for the composition of a base business model structure, hence leading to a first proposal integrating the popular BSC and BMC reference models. Nevertheless, due to the diverse concepts involved and the use of different perspectives it presents itself as a complex solution, especially for novice or not so experienced users.

In order to handle these issues, an evolution of the proposal was presented integrating it in a three-dimensional, cube-like format, with a central, strong focus on the role of the BSC as a link between organizations strategy and operations. This is due to the positioning of the BSC at the heart of an organization strategy management, being closely related to its business model. Its interrelation with functional and nonfunctional perspectives, as business processes and business goals and rules, allows for an expanded three-dimensional view and control over the entire strategy and vision of an organization.

With the use of modeling features, as well as of popular referentials inside the area for this proposal, it allows for further development of the proposed solution and leaves an open door to future connections to other points of interest. Also, its 'standard-oriented' specification, not so high-level neither low-level, intends to allow support and communication between managerial and technical sides.

The ongoing and planned work with the PGR and BSC framework integration, further detailing and analyzing the PGR-BSC Cube solution, while counting with the support of SPEM and BMM specifications, presents promising directions where to further develop, test, evaluate and evolve this research work.

The development of an assisting support tool to explore and process the different perspectives presents itself as an interesting option for the initial evaluation and testing of our proposal. The use of a rapid-development platform (OutSystems Studio), suited to the initial development in the form of a prototype, allowed shortening paths and deadlines for this solution.

CHAPTER 6

DERIVATION OF PGR-BASED SERVICE-ORIENTED ARCHITECTURES

Summary: Service-Oriented Architectures aim to address the complex challenges faced nowadays with technology, information and strategy, in order to support flexible business processes that link the value of services with the needs from the customer perspective. Requirements elicitation, evermore grounded on quality issues, allow deriving candidate architectures aligned with a business and strategic view of the system. In order to fully support this, a three-dimensional proposal for generating the quality information associated to architectural services from the elicitation of PGR-based requirements is presented. This solution is supported in a metamodel for its representation and a tailorable method for its manipulation, and, once more, associated to the development of a tool support within the OutSystems Studio environment, with its entities structure, screen flows and form captions.

CHAPTER TABLE OF CONTENTS

PUBLICATIONS:

- A SoaML Approach for Derivation of a Process-Oriented Logical Architecture from Use Cases (**IESS'15, Springer**)
- Generating a Logical Architecture Services' Quality Characteristics Aligned with its Business Requirements (**WorldCIST'15, Springer**)
- Alignment between the Business Requirements and Quality Characteristics of a Logical Architecture (**IESS'16, Springer**)

—— This page intentionally left blank ——

CHAPTER 6

DERIVATION OF PGR-BASED SERVICE-ORIENTED ARCHITECTURES

"It is the obvious which is so difficult to see most of the time. People say 'It's as plain as the nose on your face'. But how much of the nose on your face can you see, unless someone holds a mirror up to you?"

– Isaac Asimov, **I, Robot** (1950)

6.1 INTRODUCTION

Referring back to Chapter 4, a second view on alignment in the V-Model development process is the transformation between requirements and architecture, where the Four-Step-Rule-Set (4SRS) method plays a crucial role. Following our proposed evolution for the requirements elicitation in handling the processes, goals and rules (PGR) trio, it is thus mandatory to evolve and adapt the corresponding mechanisms on the 4SRS method, in order to have a continued, integrated and coherent solution.

First, our study embraces designing logical architectures for cloud computing environments, which can be a complex endeavor moreover when facing ill-defined contexts or insufficient inputs to requirements elicitation. As existing solutions are no longer enough to embrace challenges brought by complex scenarios and multi-stakeholder realities, for example in Ambient Assisted Living (AAL) ecosystems, new concepts and cross-domain solutions emerge. These tackle those problems by connecting evermore the world of requirements and architectures, as well as of business and technology, through service-oriented approaches.

This due, we propose to extend the 4SRS method, which has proven successful in generating a proper candidate logical architecture for an information system in ill-defined contexts, to a SOA approach for greater business integration, flexibility and agility. By using the Service oriented architecture Modeling Language (SoaML) to develop and represent the elements in the generated architecture, we propose the new 4SRS-SoaML method. Accordingly, we present the result of a demonstration project based in an industrial live setting, where the 4SRS-SoaML reshaped method was applied for generating the architectural participants and respective channels of services and requests.

Also, as the derivation of logical architectures has been largely focused on the elicitation of functional requirements, disregarding the nonfunctional ones, relevant business requirements content is not reflected in the information system architectural solution, which lowers its quality. Although the relation between requirements and architecture is a crucial part of an information system, being one of the main challenges for its successful development, traditional projects keep focusing on the connection of functional requirements with architecture components having a tendency to ignore quality concerns.

As the quality attributes of a system support its architecture high level structure and behavior, also being highly related to its early nonfunctional requirements, aligning these two realities is a pressing need. Although research has recently been approaching this issue much is left to do, especially regarding the alignment of the business-side requirements with the information system-side logical architecture components.

Following our earlier proposed metamodel for relating PGR-based requirements elicited from business (Chapter 4) and the now described 4SRS-SoaML method for the derivation of a logical architecture in SOA environments, we then extend our work by generating the quality information associated to architectural services from business requirements. This solution for aligning business requirements with services quality characteristics, by derivation of a logical architecture, extends the PGR metamodel to include the architectural services and associated quality characteristics. As a result, it contributes to the improved alignment and traceability between the use-cases problem-set and the logical architecture's components solution.

Additionally, the dual specification of a metamodel and method supporting a three-dimensional approach for handling the alignment of the quality issues between requirements and architecture is proposed. By taking advantage of a cube structure and method definition within an approach specified under the Software and Systems Process Engineering Metamodel (SPEM), which is adaptable to model variations, our proposal constitutes an improved, multi-project, user-friendly and tailorable solution. This is supported according to its continued use within our research group and its added value in comparison with the previously existing approach. Accordingly, an illustration of its use and an accompanying discussion are presented, and further complemented with the development of a support tool, in the form of a prototype, for the cube manipulation. Finally some conclusions are drawn for this chapter.

6.2 A Method for the Derivation of SoaML-based Architectures

As our digital life and work increases the need for more specific services, nowadays cloud oriented, the pressure is on for software architectures to lead the development of these services. However, designing logical architectures for cloud computing environments can be a complex endeavor, moreover when facing ill-defined contexts or insufficient inputs to requirements elicitation.

Building systems with well-defined component interfaces, ready for effective reuse and maintenance, requires a robust and realistic development process, where both requirements engineers and system architects work concurrently and iteratively to describe and align the artifacts they produce. In fact, candidate architectures constrain designers from meeting particular requirements and the choice of requirements influence the architecture that designers select or develop. This arises in an iterative process that produces progressively more detailed requirements and design specifications [Nuseibeh, 2001].

Also, continued research on service-oriented technologies and management brings new conceptual frameworks and theoretical perspectives to firms that adopt and implement them, helping to address the major challenges faced with technology, information and strategy. Recent growing research supports companies on delivering new, more flexible business processes that harness the value of the services approach from a customer's perspective, which are needed to leverage technology in response for greater business integration, flexibility and agility [Demirkan *et al.*, 2008].

The 4SRS method has proven successful in generating a proper candidate logical architecture, in ill-defined contexts and with multiple stakeholders, by eliciting and managing requirements from a process-perspective in an AAL project [Ferreira *et al.*, 2014]. Nevertheless it failed to provide a more clear business and strategic view of the system with its large and complex architecture, and its oversimplified architectural elements.

Due to the increased complexity and challenges posed by new realities and the emergence of more robust, cross-domain, solutions, we felt the need to improve the 4SRS method in order to answer these issues. In this sense, our proposal is to extend the 4SRS method by adapting its steps while taking advantage of the SoaML concepts, in order to build a stronger, Business-Information System aligned solution.

We tested the new method on a demonstration project, in an AAL industrial live setting, where the method was applied by generating the architectural participants and respective channels of services and requests, and analyze its results comparatively to a previous existing solution. Next we present the description of the changes proposed in the steps involved in the new 4SRS-SoaML method highlighting the differences to the conventional approach. Accordingly, a demonstration of the proposed method, applied in an AAL project is presented and a brief analysis on the ongoing research and preliminary results obtained is performed.

6.2.1 Design of the 4SRS-SoaML Method

Although the 4SRS method has proven successful in generating a proper candidate logical architecture, in ill-defined contexts and with multiple stakeholders, it failed to provide a more clear business and strategic view of the system. Its large, complex architecture and oversimplified architectural elements call for more full-bodied, cross-domain solutions, with in-depth detailed elements, as the ones found in SOA, namely through the definition of different levels of architectures, composed of services and requests, and their properties, as specified by SoaML.

Although SoaML allows building a complete service architecture, for now our work will focus solely on using system requirements to identify participants (P) and its associated requests and services, enabling the design of a logical architecture based only on these participants. The process-level 4SRS method seems capable of supporting such design by maintaining its main steps but changing some rationale.

Here, our purpose is to focus on how SOA concerns – by using the SoaML notation – are treated during the execution of the four steps of the method and focus on the main differences to the conventional method execution, rather than exhaustingly describing the full method's steps and micro-steps. This application differs from the conventional approach by defining a set of added rules that must be observed when reasoning about the execution of the method steps.

The starting point of our approach is the elicitation of UC using refinement techniques (Fig. 6.1a). The leaf UC of this model are used as inputs for the 4SRS-SoaML method execution (Fig. 6.1b), where through it, participants (P instead of the conventional AE) are derived, as well as their respective requests, services and properties (Fig. 6.1c). Participants are used to define the service providers and consumers in a system, where they may play the role of service provider, consumer or both, communicating between service/request channels as described in SoaML. After aggregating participants in super-participants (SP), later described in Step 3 of the method, the main service/request channels are identified and ready to be thoroughly specified (Fig. 6.1d).

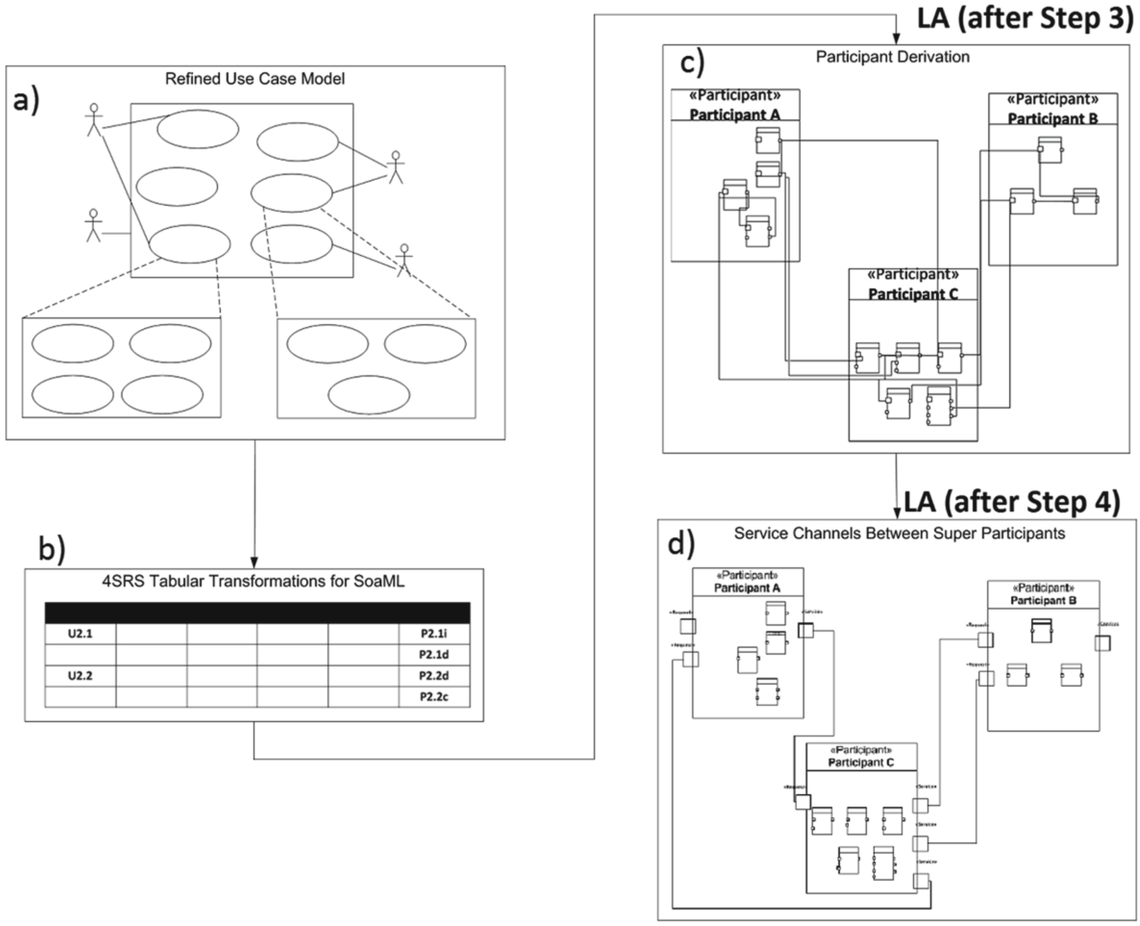


Fig. 6.1: From use-cases to service/request channels

Step 1 – Participant Creation

Now called Participant Creation, instead of AE Creation, wherein the conventional method automatically derived at least three types of AE classified as interface, data or control (referred using i-type, d-type, or c-type stereotypes). Within this new approach, the same reasoning is used but rather to automatically derive at least three types of P. Other small difference within this step regards the prefix reference that identifies the derived participant, “P” instead of the conventional process-level 4SRS “AE”, with the suffix regarding the given type having no changes (e.g., {P1.1.i} instead of {AE1.1.i}, where 1.1 refers to the UC number and i to an interface type).

Step 2 – Participant Elimination

Composed by the equivalent eight micro-steps of the conventional method, the "new" issues for eliciting a SoaML participant-based logical architecture are dealt with in nonautomatic micro-steps (only micro-steps ii and vi are automatic). The main differences on reasoning are in micro-steps iv, v and viii (Participant Description, Participant Representation and Participant Specification, respectively). The remaining micro-steps (i, iii and vii) are executed applying the same reasoning as in the conventional method execution, the only difference is to deal with P rather than AE.

Micro-step i – Use-case Classification: in this step, each use-case is classified according to the nature of its P, previously created in step 1. The nature of a P is defined according to the suffix it was tagged with. This classification is represented in the 2nd column of the 4SRS table, whereas the 1st column regards the P identification (Fig. 6.2). In the process-level perspective more than one of each P type can be generated according to the textual description and in the model of the use-case. Each P type must be interpreted as an i-type (referring to interface, represents process' interfaces with users, software or other processes which interact with other P external to itself), a c-type (referring to control, represents a process focusing on decision making and such decision must have a computational support) or a d-type (referring to generic decision repositories not computationally supported which stores information).

	Step 2 - participant elimination			
	2i - use case classification	2ii - local elimination	2iii - participant naming	2iv - participant description
{0b.1.2}	ci			
{P0b.1.2.c}		T	Health Monitoring Decisions	Short Description: Makes decisions on how the measured information from {P0b.1.2.i} is used within the AAL4ALL Node. The information can be used by the platform for preventing abnormalities... Requests: Monitoring Configurations (0b.1.1.i.s) Receive Current Vital Signs (0b.1.2.i.s) User Validation (0b.6.1.1.c.s) Services: Health Monitoring Decisions (0b.1.2.c.s) Properties/Resources: Configurations User's Activities Vital Signs Values
{P0b.1.2.i}		T	Receive Current Vital Signs Information	Short Description: Receives the current values for vital signs (e.g., blood pressure, heart rate, etc.) measured by the health monitoring devices. The information is published. Requests: Services: Receive Current Vital Signs (0b.1.2.i.s) Properties/Resources:

Fig. 6.2: Micro-steps i to iv of Step 2 of the 4SRS-SoaML method

Micro-step ii – Local Elimination: refers to determining which P must be eliminated in the context of a use-case, guaranteeing its full representation. This is required since micro-step 2i disregards any representativeness concerns. There are cases when there is an explicit place for a d-type P and it is immediately eliminated. Reasons for this are due to the process-level perspective, as there is no need for certain types of decision repositories that only regard information for the final product and not the process.

Micro-step iii – Participant Naming: in this micro-step (4th column in Fig. 6.2), the P that survived the previous micro-step are given a name. The name must reflect the role of the P within the entire use-case and also its type (c, d or i), in order to semantically give hints on what it represents and not just copy the original use-case name.

Micro-step iv – Participant Description: regarding micro-step iv, our proposal provides an extended structure for the required description of the P's role for the service behavior. This information has as input the textual description of the original use-case from the UC model. The P description includes an overall description of its behavior (as in the conventional method), where there must be defined how it acts to provide or consume a given service/request, if it has any properties, etc. (Fig. 6.2). Additionally, from the use-case's textual description, it is possible to identify some requests, services and properties.

Micro-step v – Participant Representation: its purpose remains from the conventional method, i.e., to identify a participant whose role can be represented by another participant from the global architecture. This micro-step signals P that can be eliminated without losing coherence within the architecture. The decision to identify if a participant represents another participant, or if it can be represented in the system by another one, is based solely on their overall description. If any P under analysis (to be represented or not by others) have different requests or services regarding the participant that will for now on represent them, that does not allow to directly infer the representation of these P into a single participant at this moment. Rather, the mixing of requests, services and properties regarding these given P are only dealt with in micro-step viii.

Micro-step vi – Global Elimination: represented in the 4th column of Fig. 6.3 and similar to micro-step ii, it identifies which P must be eliminated in the context of the global model. Its execution is automatic, where the P that is represented by itself or represents other P is maintained, the rest (i.e., P that are represented by other P) are eliminated.

Micro-step vii – Participant Renaming: recorded in the 5th column of Fig. 6.3, P that have not been eliminated in micro-step vi are renamed. In cases where the P under analysis results of the representation of more than one P the new name must reflect its global execution in the project context.

Micro-step viii – Participant Specification: this is still similar to micro-step iv, since it intends to describe P that, in micro-step v, were considered to represent other P without losing any information (role and objective of the participant, requests, services and properties), and were meanwhile eliminated in micro-step vi. Thus, for defining the Participant Specification, it must be taken in consideration that only redundant overall description of the behavior of the participant, requests, services and properties are eliminated, and that any of these fields that differ from the P to be represented should be incorporated in this "new" specification that represents them (Fig. 6.3).

Step 3 – Participant Aggregation

In Step 3, the enduring P (those that were maintained after executing the entire Step 2), for which there is an advantage in being treated in a unified process, should give origin to aggregations of semantically consistent P. In the conventional method (both process and product-level) this step outputs packages for aggregating AE, objects or components, while in the new approach this step outputs SP instead. A SP represents a group of assigned P which may or may not represent a software product from the services ecosystem in question. The aggregation performed can also reflect identified product classes or other aggregated types of service suggested by reference architectures or models. The notational representation of SP in SoaML remains the same as for P, only viewed one level up.

	2v - participant				
	represented by	represents	2vi - global elimination	2vii - participant renaming	2viii - participant and service specification
{0b.1.2}					
{P0b.1.2.c}	**{P0b.1.2.c}**	**{P0b.3.3.1.c}**	T	Health Monitoring Decisions	Short Description: Makes decisions on how the measured information from **{P0b.1.2.i}** is used within the AAL4ALL Node. The information can be used by the platform for preventing abnormalities in User's wellbeing while he is at home (routines, sport exercises, during sleep etc.). The Information is also used by the Formal Caretaker (Doctor) ... Requests: Monitoring Configurations (0b.1.1.i.s) Receive Current Info (0b.1.2.i.s) User Validation (0b.6.1.1.c.s) Services: Health Monitoring Decisions (0b.1.2.c.s) Properties/Resources: Configurations User's Activities Vital Signs Values
{P0b.1.2.d}					
{P0b.1.2.i}	**{P0b.1.2.i}**	**{P0b.1.3.i}** **{P0b.1.4.i}** **{P0b.3.3.1.i}**	T	Receive Current Information	Short Description: This Participant is responsible for receiving all monitoring-related information from the several monitoring devices. Receives the current values for vital signs, the current information regarding equipment usage and User´s steps. The information is published in the AAL MQ. ... Requests: Remind Meds (0b.3.2.1.c.s) Remind Consults (0b.3.3.3.c.s) Remind Events (0b.4.4.1.2.c.s) Services: Receive Current Vital Signs (0b.1.2.i.s1) Receive Equipment Information (0b.1.2.i.s2) Receive Routines Information (0b.1.2.i.s3) Properties/Resources:

Fig. 6.3: Micro-steps v to viii of Step 2 of the 4SRS-SoaML method

In this new approach, there is the addition of dividing this step in two new micro-steps. Besides the conventional identification and naming of the SP, with the inclusion of the P, here performed in micro-step i, its characterization, similar to a description of a product class, is performed in micro-step ii (Fig. 6.4). This last micro-step can be performed by defining the descriptions in a structural way, following guidelines as, for instance, the European Telecommunications Standards Institute (ETSI), namely defining the SP description by: Name, Brief Description, Kind, Interfaces, State Variables, Requests, Services, Realization options, Prerequisites and Non-functional requirements.

Micro-step i – Super-participant Naming: performs the conventional identification and naming of the super-participant, according to the aggregation service provided.

Micro-step ii – Super-participant Description: fills in a structure for the required description of the SP role for the aggregated service behavior. This information has as input the textual descriptions of the originating P from the previous step.

Step 4 – Participant Associations

Regarding Step 4 on the conventional method (both process- and product-level), associations between AE, objects and components, respectively, are derived based on the AE objects or components originated by the same use-case or based on textual descriptions of UC. In our new approach, this step formalizes the requests and services that were previously identified in micro-steps iv and viii of Step 2, but only those regarding the P that were not aggregated in the same SP in Step 3, i.e., belonging to a different one. Our purpose in this step is to generate the main communication channels between the different SP by exposing their services and requests in the final architecture.

Micro-step i – Direct Associations: deriving from P originated by the same use-case, these associations are depicted from the classification given in the method micro-step I of step 2.

Micro-step ii – Use-case Model Associations: can be inferred from the textual descriptions of use-cases, that is, when a use-case description refers implicitly or explicitly to another use-case, the associations inferred imply that the use-cases are connected.

6.2.2 Demonstration Case

The AAL4ALL project[42] arises from the need to create an oriented national market to products and services for AAL given the trend of an aging population and the need to respond positively to the increasing availability of better health care and wellness. At the same time the concept of Cloud Computing appears in the AAL4ALL domain with the opportunities for developing new products and services that can be available on the Internet. This project presents an idea for an answer through the development of an ecosystem of products and services for AAL associated to a business model and validated through large scale trial.

AAL refers to electronic environments that are sensitive and responsive to the presence of people and provide assistive prepositions for maintaining their lifestyle. It is primarily concerned with the individual in its immediate environment by offering user-friendly interfaces for all sorts of equipment at home and outside, taking into account that, sometime in their life, people have impairments in

[42] http://www.aal4all.org/

vision, hearing, mobility or dexterity [Pieper, Antona, & Cortes, 2011]. These AAL solutions intend to monitor and facilitate health, safety and well-being of individuals in specific scenarios such as within their home, in mobility, in care centers, at work and even during recreational activities, promoting independency, mobility, safety and social contact through increased communication, inclusion and participation, using ICT [Magjarevic, 2007].

	Step 3 - packaging & aggregation		Step 4 - participant association
	3i - super participant identification & naming	3ii - super participant description	
{0b.1.2} Check			
{P0b.1.2.c}	{SP2} Home Gateway	**Description:** central node where different services (provided by several service providers) that will compose the AAL4ALL ecosystem, are aggregated, ... **Kind:** Software Application **Interfaces:** AAL Devices Cloud; Events and Transports Private Cloud; Health Care Private Cloud; **State Variables:** **Requests:** Monitoring Configuration Requirements; User Monitoring Configuration; Home Emergency Alert Requirements; Outdoor Emergency Alert Requirements; Health Monitoring Decisions; Equipment Monitoring Decisions; User Indoor Position; Show Routes; ... **Services:** Provider Validation; User Validation; Publish Business Info **Realization options:** Ethernet, AMQP and HTTP **Prerequisites:** **NFR's:**	
{P0b.1.2.d}			
{P0b.1.2.i}	{SP4} Sensors	**Description:** sensor devices that will compose the AAL4ALL ecosystem ... **Kind:** Device **Interfaces:** Home Gateway; **State Variables:** **Requests:** Home Emergency Alert Requirements; Outdoor Emergency Alert Requirements; Health Monitoring Decisions; Equipment Monitoring Decisions; Routines Monitoring Decisions; User Indoor Position; User Outdoor Position **Services:** Receive Current Vital Signs, Receive Equipment Information, Receive Routines Information, ... **Realization options:** Bluetooth, HTTP, WiFi, IEE 802.11, ... **Prerequisites:** **NFR's:**	{P0b.1.2.c} {P0b.1.3.c} {P0b.1.4.c}

Fig. 6.4: Steps 3 and 4 of the 4SRS-SoaML method

The AAL4ALL project aims to develop a core platform that allows an aggregation of all stakeholders systems to enable the composition of the AAL services that will be provided to end-users in both home and in mobility environment. Within it, several types of devices are installed to gather data from user environments and to provide AAL services. Therefore the AAL4ALL platform development should aim at providing a SOA and a cloud-based solution, capable of ensuring a set of services for the ecosystem. This project has defined four main Life Settings (LS): Independent Living; Health and Care in Life; Occupation in Life and Recreation in Life. A number of scenarios in each LS and their potential impact on the roadmap implementation are included as a starting point.

Another element of implementation is the identification of the main participants to be involved in the project, for idealization of the kind of needs specified in each scenario. The scenarios' descriptions, alongside the participating Personas, actors that represent users that interact with the projects IS, were elicited and gave origin to UC diagrams with the actors and their interactions by LS and Canonic Activities (CA) criteria, from a process point of view. The process-level UC allowed a better understanding of the projects IS related activities.

Using this information, it was possible to characterize the necessary UC for representing the high level processes related to the project. Each UC has a textual description allowing process requirements to be captured and described. It was decided to use two orthogonal views of the system under analysis: the {0a} LS and the {0b} CA (Fig. 6.5), which existence in user requirements higher abstraction level has enabled to cope with the inherent complexity when dealing with both.

The process-level logical architecture of the project ecosystem has previously been developed recurring to the conventional 4SRS method [Santos *et al.*, 2012]. That perspective allowed capturing the intentionality's presented in the desired activities that the ecosystem sustains. However, the need for a clearer business and strategic view of the system, by identifying the communication channels between the products and services within the AAL4ALL ecosystem, raised the need for deriving a logical architecture now based in a SoaML notation. This logical architecture should allow depicting a set of services channels between the participants and, in a higher view, the super-participants.

An excerpt of the demonstration case, throughout the four steps of the method execution, is included in the previous section (Figs. 6.2, 6.3, 6.4). Namely, the derivation of use-case '{0b.1.2} Check Health Values', until participants in super-participants '{SP2} Home Gateway' and '{SP4} Sensors' (Fig. 6.4) is presented throughout the steps execution and this last in the zoomed area of the process-level logical architecture for the AAL4ALL project (Fig. 6.6), after the execution of the 4SRS-SoaML method for deriving a logical architecture. Also depicted in the zoomed area is a subset of the architecture, where the super-participant '{SP4} Sensors' and its associated requests and services are presented, with the rest of the super-participants that compose the remainder of the logical architecture in the background.

The execution of Step 3 in the AAL4ALL project was based on the product classes identified in [Santos *et al.*, 2012], where the logical architecture was mapped and a new iteration of the architecture arose by following the Continua reference model. The identified super-participants in the AAL4ALL projects were the following: '{SP1} AALMQ' (cloud platform for service orchestration); '{SP2} Home Gateway' (home and mobile application hosting devices); '{SP3} Local System' (home private cloud); '{SP4} Sensors' (PAN/LAN devices, either sensors or actuators); and '{SP5} External systems' (other public or private health clouds). The super-participants of the AAL4ALL logical architecture (Fig. 6.6) regard the aggregated participants that relate to the functionalities elicited from the orthogonal view of the CA and LS.

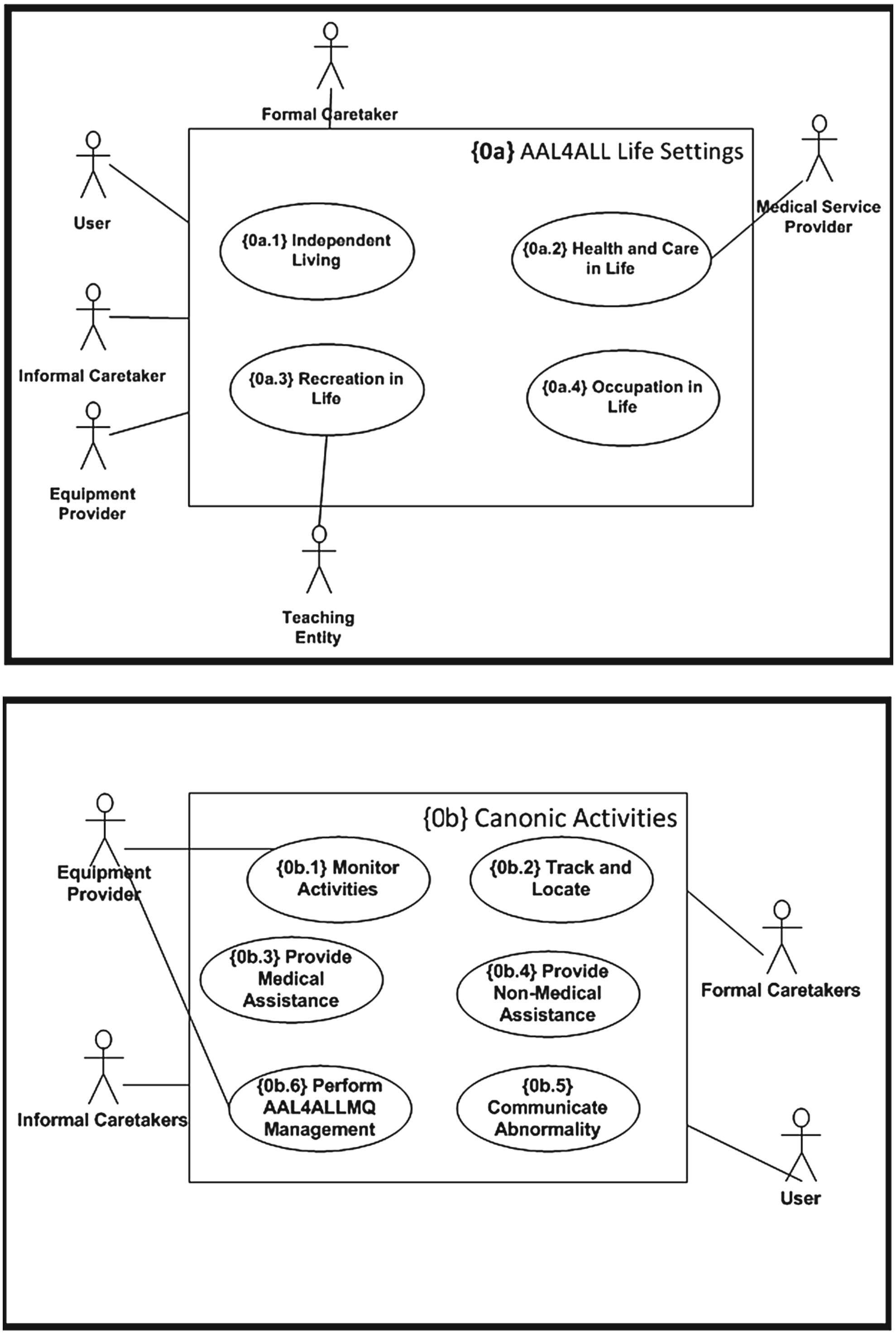


Fig. 6.5: The use-case model in two orthogonal views

The adoption of the 4SRS-SoaML method allowed the SoaML architectural model composed by participants, requests and services, representing the system requirements, to be derived from and aligned with the initial user requirements. As these result in an identification of elements in a non *ad hoc* approach, they are less prone to errors of design. An AAL4ALL logical architecture (Fig. 6.6) also shows some of the complexity regarding the project ecosystem that justified the adoption of process-level in order to support the elicitation efforts and the definition of the services channels.

A complete listing of this case study's deliverables, apart from the excerpts here reproduced, is out of scope for this document due to their extension, and also to the fact that this study is more focused on analyzing the process itself and not so much the results obtained. Nevertheless, Appendix B contains more information regarding the AAL4ALL project and the tools it used.

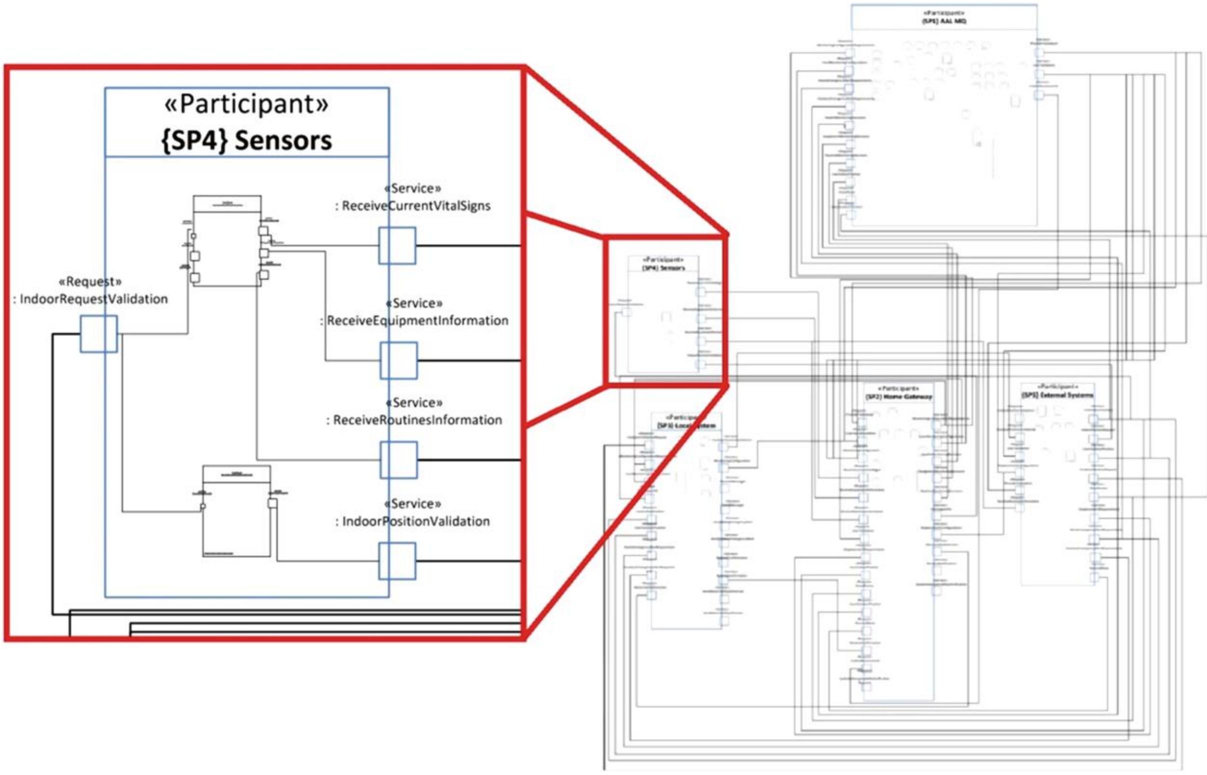


Fig. 6.6: Subset of the AAL4ALL logical architecture using SoaML

6.2.3 Discussion

In this demonstration case, the SoaML modeling language was used to support process-level logical architectures and, as such, to provide a baseline modeling language for specification of any services within a service-oriented environment, which includes cloud-based services. This work demonstrated that the SoaML profile supports the range of modeling requirements of the logical architecture, including the specifications of systems of services, individual service interfaces and service implementations.

According to our experience, at the end of the requirements elicitation phase (considering the traditional development macro-processes Analysis, Design, Implementation, Test and Deployment) implementation teams experience difficulties interpreting and implementing the generated conventional process-level logical architecture, composed by AE. The purpose of this architecture artifact is to provide information regarding the software system functionalities, regardless of the implementation techniques and technologies the team uses.

This is due to the fact that functional requirements are elicited, but technical decisions regarding messaging, protocols, amongst others are lost along the way, being unclear at this point. The authors believe that SoaML notation is more adequate for relating business and system information, leveraging the business requirements and transporting them to implementation phase. By eliciting P and its associated requests, services and properties, instead of simple AE, the provided artifacts present an added wealth of information for interpretation and implementation by the teams, when comparing to the previous 4SRS approach.

Also, associations between AE were not always understandable, since typically teams interpreted them as interactions between AE, not depicting the type of information within, as well as the course of the flows. In fact, for depicting the course, stereotyped sequence diagrams had to be developed [Touzi *et al.*, 2009]. These issues are tackled in our approach by explicitly defining P's requests and services. The combination of the 4SRS method with the SoaML language helps to strengthen the associations in terms of services and requests channels across the architectural elements, either P or SP.

The previous architecture was composed by 110 AE and ours was composed by the same amount of P, since they have a somewhat direct correlation. However:

- in the 4SRS-SoaML architecture, there are 171 requests and 143 services, which results in an addition of 61 requests and of 33 services.
- in the previous architecture there were depicted 159 associations, i.e., communication flows of any kind between the AE, in our architecture there are 139 associations, i.e., services that are requests between P.

This means that this new approach was indeed able to provide more detailed information regarding the services provided, with the excess in number of services and requests discovered. Additionally, the architectural representation was also improved due to the decrease number of associations, providing a clearer depiction of the architecture and aggregating the main communication channels information.

As previously stated, the authors believe that using the SoaML notation at the requirements elicitation phase is more useful for implementation teams since it allows including more information for the future system to be implemented, like technical data regarding the software services. Within our approach it was possible to early identify requests and services directly from the use-case's textual descriptions.

The step 3 of the 4SRS-SoaML method allowed identifying possible aggregations of semantically consistent P into SP, according to the proposal of the intended system. Somehow in this step, the logical architecture can be divided into subgroups according to the ultimate goals of the desired system, in terms of functional groups and product categorizations.

Changes made in step 4, where only requests and services between the SP are considered, are especially useful for depicting the service channels required for communications between different products that will be integrated in the future. In cases of software implementations made by multiple and distributed teams, each nominated to implement one of the SP, the service channels hidden inside the same product are no longer relevant in the overall project management.

Within our approach to elicit a SoaML architecture, we derived participants with their constituent requests, services and properties. Albeit complete Service Choreographies are depicted by RUP's Basic Flow and Alternative Flow information from the UC's textual descriptions, only generic information was derived in this work, so the model derivation has several points to be improved. In the future we intend to include in this approach the derivation of other SoaML diagrams like Service Contracts, Service Architectures and Interfaces.

Here, we derived a candidate logical architecture in an ill-defined context by executing a 4SRS method that was tailored for SoaML diagrams, which constitutes one of the steps in the V-Model [Ferreira *et al.*, 2014] approach, within other models successive derivations. This approach is composed by UML models, like use-cases, stereotyped sequence diagrams, logical architectures, amongst others. As future work, we intend to use SoaML notation throughout the entire V-Model, in order to obtain information regarding services until the product-level side of the requirements elicitation phase.

Finally, in what specifically regards the 4SRS-SoaML method, we intend to improve these service channels with business information, usually hidden in nonfunctional requirements, as business goals and rules. In order to do so, we intend to include the processing of this information within our approach, supported on the OMG Business Motivation Model (BMM) link with SoaML. This would strengthen the channels business information, associated to the connection between the requests and services of P and SP, so supporting the quality characteristics of the derived service-oriented logical architecture.

6.3 Three-dimensional Alignment of Requirements and Architectures

In our modern society, delivering a working product is no longer enough, it needs to have quality too. The same applies to information systems architectures, besides presenting a working solution analysts need to justify its components while relating and tracing them back to business requirements [Winkler & von Pilgrim, 2010]. Notwithstanding, as the derivation of logical architectures has been largely focused on the elicitation of functional requirements, disregarding the nonfunctional ones [Chung *et al.*, 2009], much of the business requirements content is not reflected in the information system architectural solution, so lowering its quality.

Although recent research has been approaching these issues, namely with SOA [O'Brien, Merson, & Bass, 2007], much is left to do, especially with regard to the alignment of business requirements with the logical architecture components. As architectures are the bridge between business and software-intensive system, with the quality attributes requirements driving its design, it is important to understand how to support them. Choosing and designing architectures that satisfy the functional as well as the nonfunctional or quality attribute requirements, is vital to the architectures' success.

Also, the transition from and consequent alignment between requirements and architectures has consistently been one of the main challenges during information system development, namely in the development of software solutions [Galster, Eberlein, & Moussavi, 2009]. One of its key focuses is on creating architectures that satisfy requirements and their underlying intent, while addressing software quality attributes and supporting traceability management.

Moreover, as human intervention is indispensable due to the complexity of the tasks involved, a certain level of automated support in the form of a tool is essential [Yue *et al.*, 2011]. Dependencies that exist between requirements and architecture have been referred to as the twin peaks of requirements and architecture, where one of its challenges is preventing the vulnerabilities of traditional projects that focus mainly on functionality while ignoring quality concerns [Cleland-Huang, Hanmer, Supakkul, & Mirakhorli, 2013].

Quality issues are ever present in the development and evolvability of an architecture, where the connection to the nonfunctional side of requirements raises a number of questions [Loniewski *et al.*, 2010]. These relate with problems as the management of requirements whilst simultaneously stressing the automation benefit of its transformation to architectures, or the need for more efforts to explicitly deal with requirements traceability while providing better tool support.

An adopted solution is using models in the requirements phase, serving as input for model-driven transformations and making further use of them in an automatic manner. This profits from the use of model-based techniques in the improvement of productivity, efficiency, and software development process quality and effectiveness [Breivold *et al.*, 2012]. It also strengthens the foundation of the software system and its architecture, connecting its high level structure and behavior to the quality attributes.

Our previous section work on deriving a service-oriented architecture from business requirements tries to answer some of these issues, nevertheless it is not easily operationalized by users due to the diversity of concepts involved and representations of the different perspectives. This leads us to propose an approach regarding the alignment between the quality attributes of service/request pairs in a service-oriented architecture and the functional and nonfunctional side of its originated business requirements. In this proposal, the quality architecture attributes choice and representation is defined in line with the Consortium for IT Software Quality (CISQ) Software Quality Characteristics (SQC), the service/request pairs in the form of SoaML participants, and the functional and nonfunctional requirements based on the PGR approach.

Accordingly, the relation between functional and nonfunctional requirements (PGR), and from these to architectural services/requests can be viewed as a three-dimensional reality, lifting views from the plain one-dimensional prescriptions [Buglione & Abran, 2002], while added to the comfort perceptions presented to users by cube-like structures [E. G. Hansen *et al.*, 2009]. So, recurring to the 4SRS-SoaML logical architecture derivation method presented in the last section, developed in a SOA environment while counting with PGR-based requirements, the derived service/request pairs stand as a third dimension which can then be extended with a CISQ-SQC mapping. This due, we propose a solution for the representation of the three related perspectives in a cube like structure, following those three axis, from now on named the PGR-SOA Cube.

The main objective for this cube is that it can be understood and handled by different users, for diverse visualization and processing purposes. Also, as the entire method follows a model-based approach its operationalization can be traced back and forth through the different perspectives, allowing for complete traceability and an aligned solution, while facilitating future tool development. To further support this solution, it has associated a specification of its method in SPEM, where, due to its features, the method is tailored according to the project needs or authors recommendations.

Next, we present our proposal for the PGR-SOA Cube metamodel representation of its associated cube structure and its respective tailorable method to handle their visualization and processing, both supported in the alignment between the PGR and CISQ-SQC information. Then, we demonstrate our proposal, discuss and frame it inside the related work in the area, analyze contributions, current research status and envision the future work ahead. Once again, although this thesis focus is not about usability and graphical user interfaces (GUI), the advantages of working with a tailorable method and in delivering solutions which allow for its users to take full advantage of these properties, handling much of their inner complexity, raises enough concerns to corroborate these type of work.

6.3.1 Design of the PGR-SOA Cube

Our proposal for a three-dimensional approach for a quality-based alignment between requirements and architecture is grounded in the derivation of a service-oriented logical architecture, via SoaML participants, from the functional side of the elicited business requirements for an information system, as presented in the previous section. Furthermore, there is the implicit relation and alignment of their, respectively associated, SQC of the logical architecture components with the nonfunctional side of business requirements. These are supported by the SOA paradigm and the CISQ-SQC metrics, and by a functional and nonfunctional requirements base assembled within a process, goal and rule (PGR) trio.

This approach, from now on designated as the PGR-SOA Cube, is twofold. On one side, there is a metamodel for representation of the functional processes (in the form of use-cases), the nonfunctional (with its goals and rules details), the services (in the form of SoaML participants); and the measures of the CISQ (related to the SQC). Complementarily, we detail a method to handle the generated cube-structure, with its constituent activity and associated tasks, work products and roles. This method is first specified, and then further tailored and organized, using the SPEM specification for soundness and clarity reasons.

PGR-SOA Cube Metamodel

The V-Model approach already aligns use-case, problem domain, diagrams (Fig. 6.7, item A) with the candidate, solution domain, logical architecture components (Fig. 6.7, item C), these last obtained from the first through the execution of the conventional 4SRS method (Fig. 6.7, item B). Accordingly to our latest 4SRS-SoaML method, each leaf use-case now aligns directly with one or more candidate SoaML participant in the architecture (Fig. 6.7, item D).

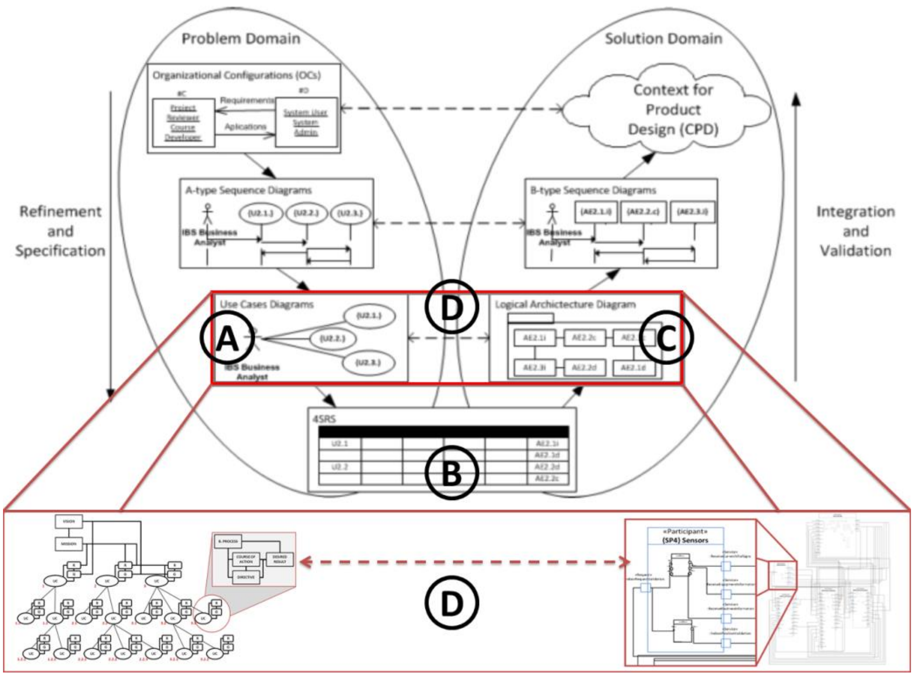


Fig. 6.7: V-Model with focus on the alignment between PGR requirements and participants

The PGR-SOA Cube follows on our previous work in deriving a services logical architecture from process use-cases, widely accepted on the functional side of requirements elicitation. These requirements combine with business goals and rules as one prominent solution on the nonfunctional counterpart in the previously proposed PGR metamodel presented in Chapter 4. Complementarily, services in the logical architecture have associated CISQ-SQC issues which can then align with the referred goals and rules, so concluding the metamodel constituents.

Following our work on relating processes (represented by use-cases), goals and rules (PGR) in the problem domain, we now aim to align and represent this business information in the architecture product related view. As previously referred, the CISQ-SQC measures present themselves as an important contribute for this issue. In order to support this alignment between business requirements in the problem domain and architectural product characteristics in the solution domain, a new approach for relating PGR, SoaML and CISQ-SQC (Fig. 6.8) is initially proposed.

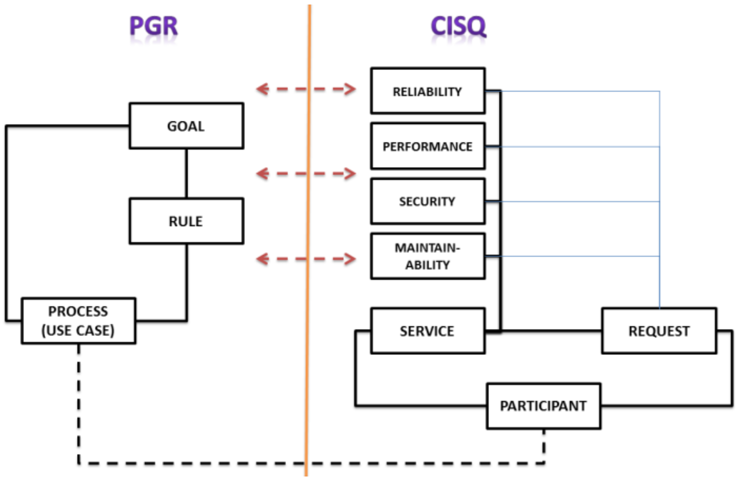


Fig. 6.8: Alignment between nonfunctional requirements and services' quality characteristics

Processes, goals and rules maintain their relation, with processes connecting to participants via the previously referred 4SRS-SoaML derivation. Each participant counts with its respective services and requests, with services connecting with their related CISQ-SQC. As each request pairs up with a service, it automatically connects with the same quality characteristics.

Moreover, the goals and rules information connects directly to the SoaML BMM container, which is associated to a Service Interface. This strengthens the linkage between the two worlds, one more business-like and the other more technology-associated. This way, a service can easily reference its business origins as well as its technological quality characteristics.

This relation has been initially explored in a method for generating the CISQ-SQC in a logical architecture from business requirements via SoaML participants. There, the CISQ-SQC elements were associated to each of the detailed individual elements of the nonfunctional part of the PGR requirements. Also, according to the proposed 4SRS-SoaML method, each leaf-level use-case requirement aligns directly with one or more candidate SoaML participants (Fig. 6.8) in the architecture.

Accordingly, in our proposed PGR-SOA Cube simplified metamodel (Fig. 6.9), the service elements are able to present themselves as a third standpoint connecting to the already existing two-dimensional realities of use-cases, and goals and rules. On the relation between these three perspectives, an associated cube-like structure is built along three axis: use-cases, goals and rules, and services (Fig. 6.10). Then, a set of CISQ-SQC measures connects directly with this triple in a cubie element.

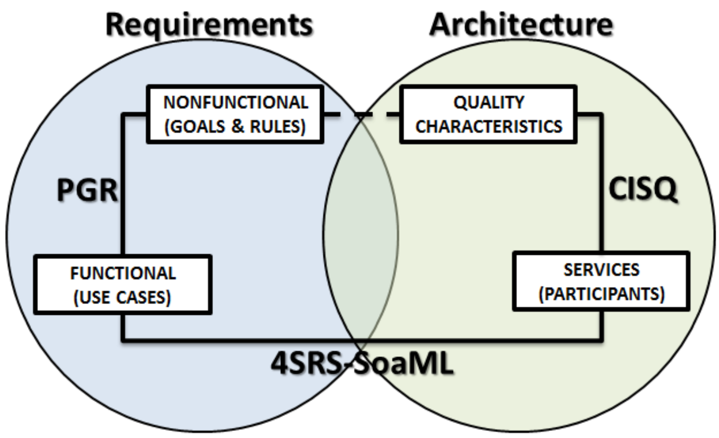


Fig. 6.9: PGR-SOA Cube simplified metamodel

The length of the process perspective is variable and depends on the number of leaf-level use-cases of the elicited information system structure (four in this example: $UC_{1.1}$, $UC_{1.2}$, $UC_{1.3}$ and $UC_{1.4}$). Regarding the services perspective, it is linked to the number of participants that are derived from the initial set of leaf-level use-cases (five in this example: SRV_1, SRV_2, SRV_3, SRV_4 and SRV_5). For the goals and rules perspective, its length is related to the three low-level constituents inherited from the BMM representation, namely:

- Objective, associated to Goal;
- Tactic, associated to Strategy;
- Business Rule, associated to Policy.

Although this last perspective follows a more fixed length status, additional goals and rules measures can be added to the PGR-SOA Cube as needed, all depending on the intended level of detail or project specificity.

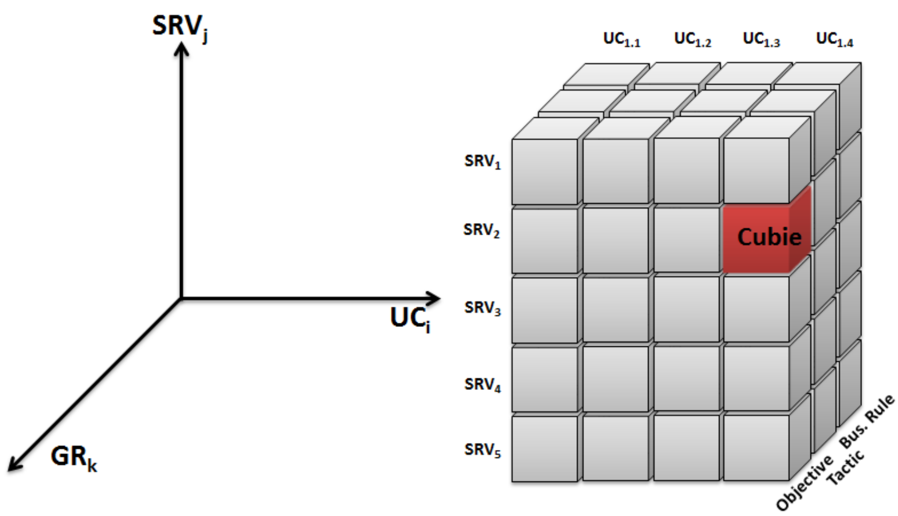


Fig. 6.10: (UC_i; SRV_j; GR_k) axis and associated cube-structure

The intersection of elements from those three perspectives represents a cubie element, as the (SRV_2, $UC_{1.4}$, Objective) example in Fig. 6.10. Inside each cubie, there is room for information regarding the system/software quality characteristic issues, in the form of statements associated with CISQ-SQC measures. To define issues for each measure, we define tuples with system/software quality characteristics (SQC) measures and statements (SQC, 'Statement'). These tuples follow on the main four CISQ-SQC measure dimensions, namely:

- Reliability;
- Performance;
- Security;
- Maintainability.

As an example, a cubie could be made out of a set of tuples {(Reliability, 'SQC statement'), (Performance, 'SQC statement'), ...} associated to the services, use-case and goal-rule dimension for which the quality issues are being defined (Fig. 6.11). These tuples are to be previously elicited from business requirements using the PGR method.

The proposed metamodel (Fig. 6.9) allows for the representation of information relating to the three perspectives and the detailed tuples of CISQ-SQC measures and statements. Also, recurring to the associated cube-structure, it is possible to adapt its visualization according to the different intentions of its diverse users (business-process analyst, systems analyst, software architect, etc.), as will be presented next.

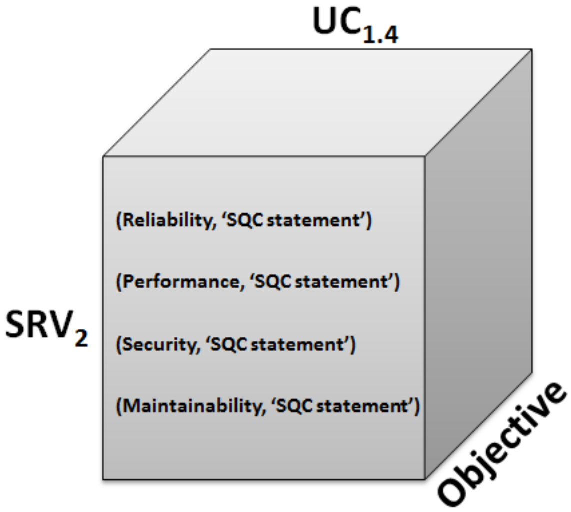


Fig. 6.11: Example of a PGR-SOA cubie, with ((O_i; T_j, BR_k), SQC) tuples

PGR-SOA Cube Method

So, to be properly integrated in the V-Model approach (Fig. 6.7), the PGR-SOA alignment method should only be applied after performing the use-case diagram elicitation (Fig. 6.7A). This includes the deduction of the goals and rules down to the leaf use-cases, followed by the execution of the 4SRS-SoaML method (Fig. 6.7B), and thus obtaining a candidate logical architecture composed of participants interconnected by services and requests (Fig. 6.7C) that can be aligned with its originating requirements.

In order to handle the information in the metamodel and the associated cube-structure, we propose the design of a generic activity (using SPEM specification), with its required detailed tasks, work products and roles. The activity is comprised of five tasks, four work products and a software architect user role (Fig. 6.12), which allows completing the alignment between the nonfunctional PGR requirements of the leaf-level use-cases with the quality characteristics of the architectural participants.

Regarding the tasks, three are associated with handling of each cube perspective and two are associated to the alignment of the services quality characteristics with SQC measures and the consequent action to perform in each cube element (cubie). Concerning the work products, three are associated to each cube perspective and one is associated to the SQC measures to align with (composing the set of cubies).

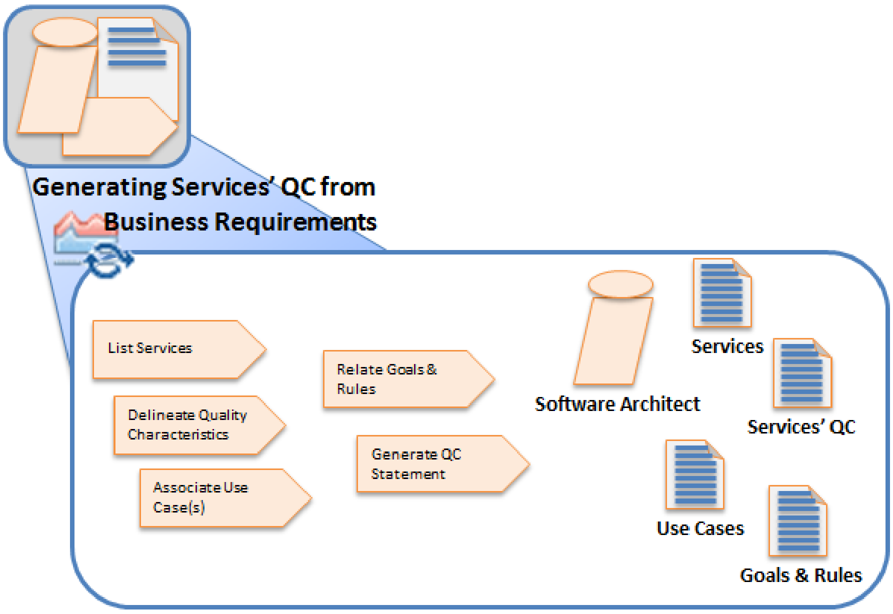


Fig. 6.12: Proposed activity with its tasks, work products and roles

So, the first three tasks involve the handling of the use-cases, the goals and rules, and the services. These are the work products that form the structure of the PGR-SOA cube, while the fourth and fifth involve the processing of the SQC measures inside each cubie, as well as the definition of the SQC statements. For the time being, the only user performing these tasks is the software architect. The detailed tasks are, respectively:

- List Services;
- Associate Use-case(s);
- Relate Goals & Rules;
- Delineate Quality Characteristics;
- Generate SQC statement from UC&GR&Srv.

Due to this specification, the activity can later be tailored in a SPEM process for operationalizing it according to the users' intentions. Each of the first three tasks in the activity can be organized and run in a cascading cycle, thus handling the elements of the corresponding perspective, for then preparing the necessary inputs or generating the intended outputs of the final tasks. The organization of the cascading tasks implies a corresponding visualization on the PGR-SOA Cube, as in Fig. 6.10, with the possibility to rotate each perspective accordingly.

That is, this type of cube-structure and associated tailorable SPEM process adapt to one another. When the cube switches in any of its three axis, the cascading order of the three first tasks reciprocally switches too, in order to conform to the desired processing sequence or to the specific visualization perspective. The two final tasks are the only ones that should not switch, as they are to be processed at the end for each cubie, detailing its CISQ based tuples.

All-in-all there are six possibilities for cube rotations, by performing different rearrangements of the three perspectives (Fig. 6.13). This cube-structure and its associated method allow handling three one-dimensional perspectives in a three-dimensional reality. Conversely, the opposite is also possible, as for example in the case of generating system documentation specifically by service or use-case (a one-dimensional reality) from the complete cube information on the three perspectives.

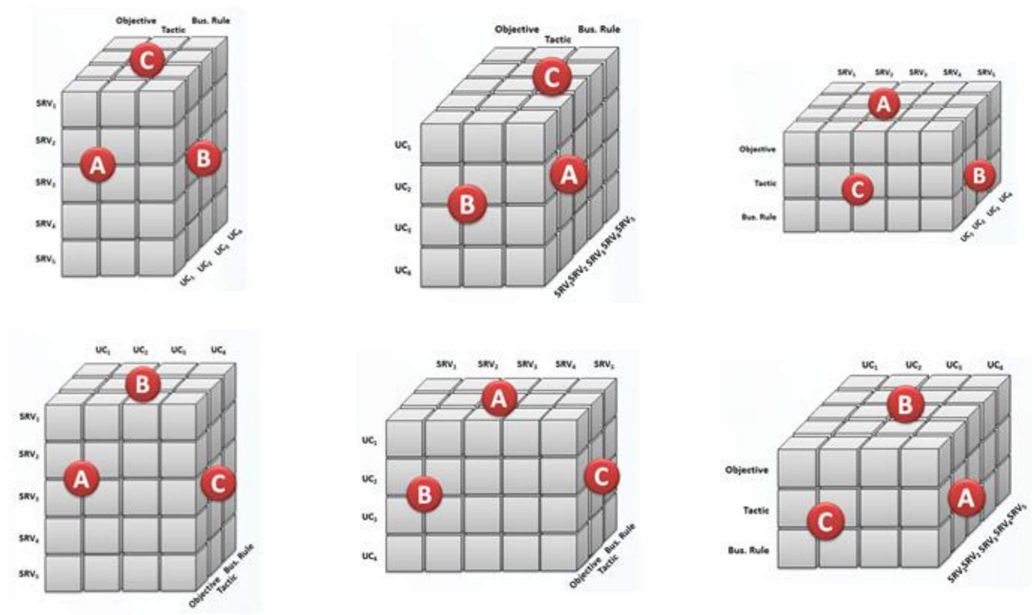


Fig. 6.13: Cube perspectives (C-Goals-Rules, B-Use-Cases, A-Services)

6.3.2 Demonstration

Having specified all the elements to be used in the method, the associated process can then be assembled and tailored to the preferences of the team that will be implementing it. The complete tailored activity 'Generating Services' QC from Business Requirements', as depicted in Fig. 6.14, receives as input the existing participants services in the derived logical architecture, is executed by a team of system architects and delivers a set of services quality characteristics.

While the three cycles inside the activity plus the relation of the CISQ-SQC measures are merely automatic, being suitable for machine processing, the specific task of defining a system/software quality characteristic statement involves all the complexity, being crucial for the successful execution of the process.

Our approach focus is on the initial aggregation and relation of the use-cases (plus the goals and rules) information to each service assured by the three cycles and going through each of the four (or more) quality characteristics, in order to search and analyze the information to generate the intended characteristic issues of the system to-be.

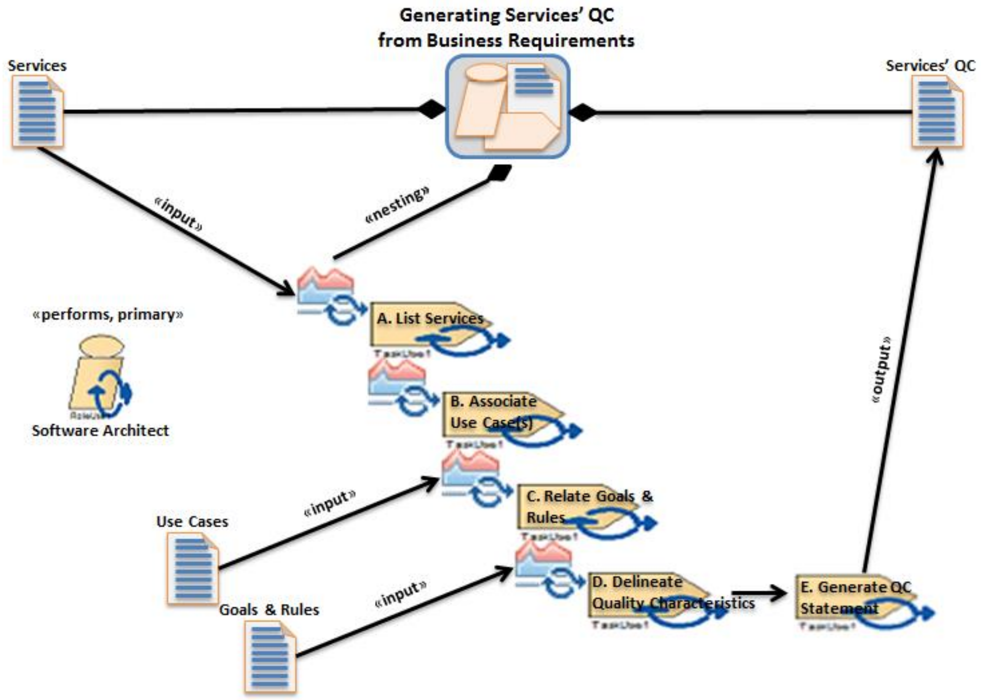


Fig. 6.14: Example tailored activity for Generating Services' QC from Business Requirements'

This interpretation and adaptation of the business-side goals and rules information directly to the logical architecture quality characteristics is, for now, the sole responsibility of the system/software architects, counting on their own heuristics and specific domain knowledge.

At the end of this process, each service counts with diverse quality characteristics issues, which later can be transformed in rules for specific system or software solutions, derived from the goals and rules associated to its originating use-cases, and so, in a totally traceable and aligned solution.

This demonstration allows exploring the possibilities of our three-dimensional, quality-based approach, the PGR-SOA Cube. By using its structures and the associated tailorable method one can obtain the same or even better results than with the previous approach, in a more user friendly way, through an improved, conceptualized, visualization of the entire method.

This activity can be performed by our method through the use of one of the six available cube-structures (Fig. 6.13) and the corresponding tailored activity (Fig. 6.14). The chosen activity guides the user through a cascading cycle in the following way:

- Select a Service;
- Select an associated Use-Case;
- Select a Goals & Rules dimension;
- Select a SQC measure;
- Define a SQC statement.

In this particular case, first the user selects a specific service then selects one of the services associated use-case(s) and finally one of the goals and rules dimensions. Having set the three perspectives, the user then selects one of the CISQ-SQC measures and performs the desired action, in this case defining a SQC statement for the current cubie accordingly.

By iterating through these three perspectives in the stated order, the user will fill each cubie with CISQ-SQC measures, statement tuples, until all the entire PGR-SOA Cube is filled-up. The order of processing for the three perspectives was tailored according to our project needs (A-B-C), but depending on the needs of other projects or users they could be easily switched, always remembering that the activity should be tailored according to the selected cube-structure.

The execution of the final task at each cubie is the sole responsibility of the system/software architect, while the other tasks can be more or less automated according to each method implementation. Filling all the cubies is not mandatory neither there is a minimum number of SQC tuples, nevertheless, it is important to fill the most part of then and detail the quality characteristics according to the schedule and detail required by the project. Moreover, in order for this task to be successful, the system/software architect must possess sound experience and knowledge regarding guidelines and heuristics on the topic of the CISQ referential.

After the execution of this activity, another possible configuration for this activity would be to fill-up the documentation for the candidate system architecture, be it by organized by service, by use-case or be CISQ-SQC measure, with the order of processing for the three perspectives being tailored according to the user preference. This activity would be entirely automatic.

6.3.3 Discussion

Research in quality requirements and architecture is still scarce, with much room for evolution in the coming years. However, there are already sound basis for the early selection of a candidate logical architecture [Losavio, Chirinos, Matteo, Lévy, & Ramdane-Cherif, 2004], considering known standards as the Unified Process (UP) and the now deprecated ISO 9126-1 quality model framework.

By prioritizing use-cases, associating quality requirements directly to them, they provide a better justification for the selection of the key use-cases relevant to the baseline architecture. Furthermore, more recently a Unified Process for Domain Analysis (UPDA) was proposed [Losavio, Matteo, & Camejo, 2014] based on Aspect and Goal orientations to deal with nonfunctional requirements (NFR), and already using the latest ISO/IEC 25010 quality standard.

Regarding the elicitation and treatment of NFR in architecture design, this is still a very fuzzy topic, as there are many transversal issues involved. Trying to integrate existing research has proven to be difficult [Ameller, Ayala, Cabot, & Franch, 2013], as some studies focus on NFR aspects while others explore software architecture issues in more depth, or even when comparison studies results do not have the necessary rigor.

Concurrently, quality requirements are becoming also a concern of both requirement engineering specialists and software architects, however, the majority of empirical studies still takes the requirements engineer analysts'/clients' perspectives [Daneva, Buglione, & Herrmann, 2013]. Even so, software architects seem to be alert in approaching the quality requirements with the same due diligence as the functional requirements, acting proactively and embracing responsibilities over them, as well as prioritizing them accordingly. Nevertheless, the quality requirements engineering area seems to still be perceived by architects as a mere social activity and not as much as a tool and method-centric activity.

On the other hand, cube structures are interesting in the way they allow for better perceptions from the users and added flexibility in the techniques to work with them. This is important to aid in the work towards reducing the gap between experts and novices, by supporting practical roadmaps, frameworks, and guidelines that can be more linear and easily taught to students, novices or analysts outside a specific domain. These are key issues, as most approaches are hard to tackle at first and require a significant level of skill and expertise from the analysts to be used effectively.

An example of using three-dimensional perspectives is the generic model termed the “Sustainability Innovation Cube”, already referred in the previous chapter 5. Another example is our proposal for the specification of a three-dimensional business model, covering the elicitation of business goals and rules from process-level use-cases, and their connection to balanced scorecard. This was explored in the previous chapter within a practical application scenario, by revisiting a project for elicitation and generation of a business model canvas, where it allowed for the use of different viewpoints to perform diverse business model transformations.

The added value of the additional CISQ-SQC measures in an existing functional and nonfunctional requirements metamodel specification, previously aligned to a service logical architecture, alongside its cube axis rotations and visualizations, and also the SPEM activity tailoring, count as an original step forward regarding existing solutions in this topic. This contributes in advancing the research knowledge, and supplies practitioners with a set of options for processing and visualizing the alignment of services’ quality characteristics in more linear ways.

The focus on the alignment between CISQ-SQC and the nonfunctional part of a system requirements elicitation allows for an added architecture quality management support, enhancing the interconnection between the business and the IS/IT realities. Likewise, existing interconnection of CISQ with both problem-side elicited functional and nonfunctional requirements, and the technical-side architecture model elements, permits widespread traceability throughout the entire solution and flexibility in applying any of the available transformation perspectives.

Lastly, the benefits of using this cube are reinforced by its continued use in projects within our research team, while the acceptance and publication of papers in relevant conferences sustain its theoretical arguments, namely within the associated research community.

6.4 Tool Support: The PGR-SOA Modeler

Continuing once more in our choice for the development of an agile prototype, available through an aPaaS solution with the OutSystems Studio, we now propose a tool support solution for the PGR-SOA Cube, from now on designated the PGR-SOA Modeler. After the initial development of a somewhat simple solution, the PGR Modeler, in Chapter 4, followed by a more complete and complex tool support, the PGR-BSC Modeler, in Chapter 5, the present proposal maintains more or less the same number of entities, web screens and operations inside events as the previous PGR-BSC Modeler.

Once more, according to our needs, we will be using mostly the Data and Interface layers, as even if the size and complexity of this particularly prototype is relatively high, there is still no need for defining high-level business processes or broad business rules. Next we detail the relevant decisions made in these two layers for the PGR-SOA Modeler prototype and additionally browse the working environment with some screenshots of the application.

6.4.1 Entities Structure

In order to implement an application to support the previously proposed metamodel, a set of corresponding entities were created, along with the necessary adaptations for a relational database. The list of entities is once more created in the Data layer of the OutSystems Studio, alongside the attributes of each entity and the automatic CRUD (Create, Read, Update, Delete) operations to perform with them (Fig. 6.15). Also included in this layer is the PGR_SOADataModel entities diagram, presenting the relations between all entities (Fig. 6.16).

As in the former PGR-BSC solution, an entity Cube was created to allow for the creation and maintenance of diverse solutions for different projects, otherwise only one cube could ever be created with this application. According to the metamodel, the use-cases processes are represented in an entity UC and the services in an entity SRV, while the goals/rules and SQC dimensions have been aggregated in two entities each.

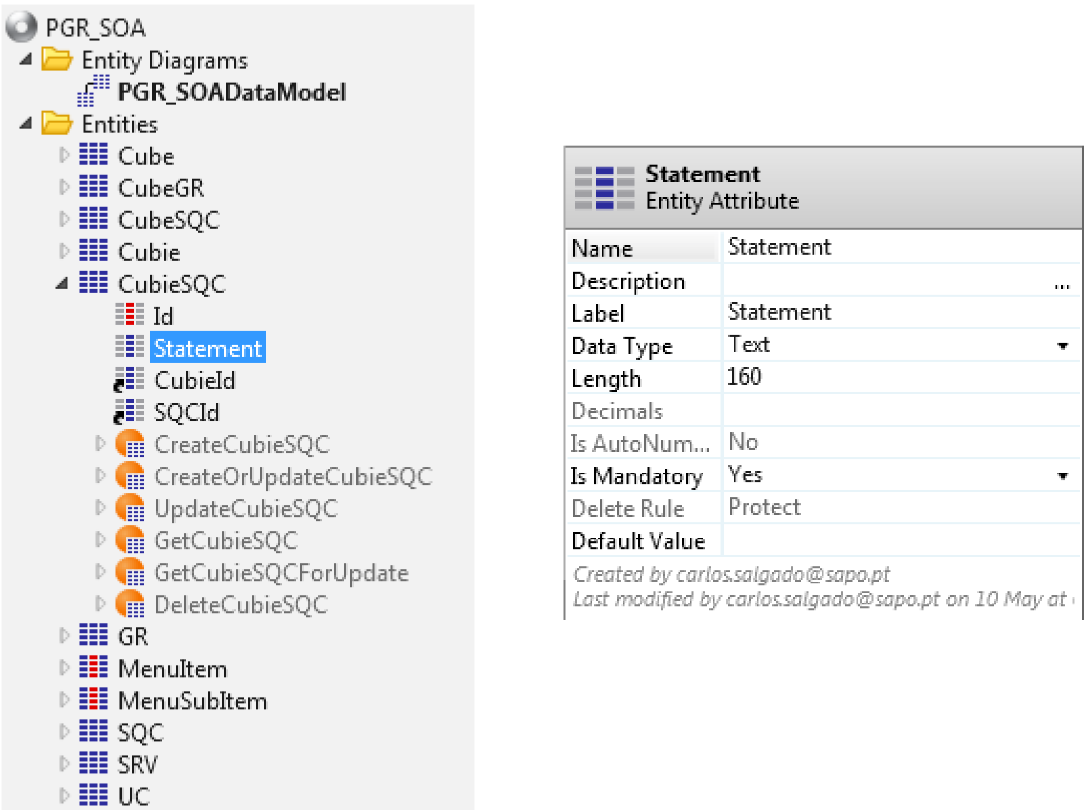


Fig. 6.15: Data Layer with the list of entities and detail for the Statement attribute of table Cubie

The entities GR and SQC allow for the base definition of any items related to these two dimensions, while the entities CubeGR and CubeSQC only associate to each cube the dimension items to be used within it. For example, we can define all the six BMM items in the entity GR, but only associate to a particular cube the bottom three, more concrete, items. As use-cases and services are directly related to a cube, they only need one specific entity each. Even though, the SRV and UC entities can also be connected since they have a relation set by the previous execution of the 4SRS-SoaML method.

Accordingly, the definition of every single element inside the cube is represented in the entity Cubie. Besides referencing many of the previously referred entities (Cube, UC, SRV and GR), it includes a set of statements describing the associated SQC propositions, for a determined goal/rule dimension, and a specific service and use-case. In order to be able to set the associated SQC statements of each cubie element, two added entities are defined, the SQC and the CubieSQC. This last entity serves only for generalization purposes of the CISQ measures.

As can be seen in Fig. 6.16, the Cube entity possesses external connections to the CubeGR, CubeSQC, SRV and UC entities (through the CubeId attribute), while the Cubie entity relates to those same entities but referencing their respective ids (GRId, SQCId and UCId), added to a general descriptive attribute Statement. The complete diagram for the PGR-SOA data model is presented in Fig. 6.16, where all the relations and dependencies between the different entities can be easily visualized. Again contrasting with the PGR Modeler and in line with the PGR-BSC Modeler, the use-case entity cannot reference itself, as in this case we are only working with the leaf, bottom-level, use-cases elicited for the information system.

Note also that although the CubeID attribute in the Cubie entity is not structurally necessary (it could be obtained through any of the SRV or UC entities), its presence eases development steps ahead. The quantity of entities in the PGR-SOA Modeler is inline with the ones on the PGR-BSC Modeler. The connection between SRV and UC has not been implemented yet, but is a natural sequence of the elicitation of the use-cases and the following derivation of the services.

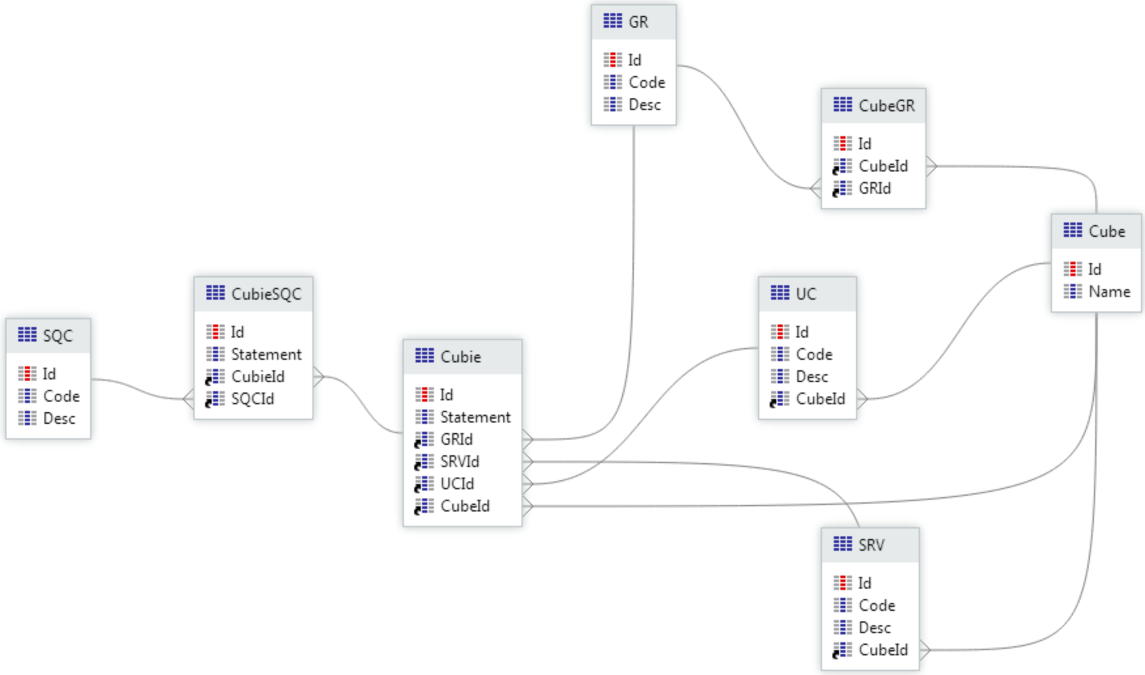


Fig. 6.16: Entities Diagram

6.4.2 Screen Flows

After finishing the entire entities structure, it is again time to design the interface screens and the associated flows connecting them. Besides that, local entry variables are added for control purposes and associated events created and/or configured for each screen. As usual, the most common of the events, already existent in every screen due to its importance, is the Preparation one, where specific business and/or technical logic is encapsulated to prepare data for the screen design. This is all possible through the Interface layer (left side of Fig. 6.17). An example of a code 'flowchart' in the Preparation event for the CubeDetail screen is also shown in the right side of Fig. 6.17.

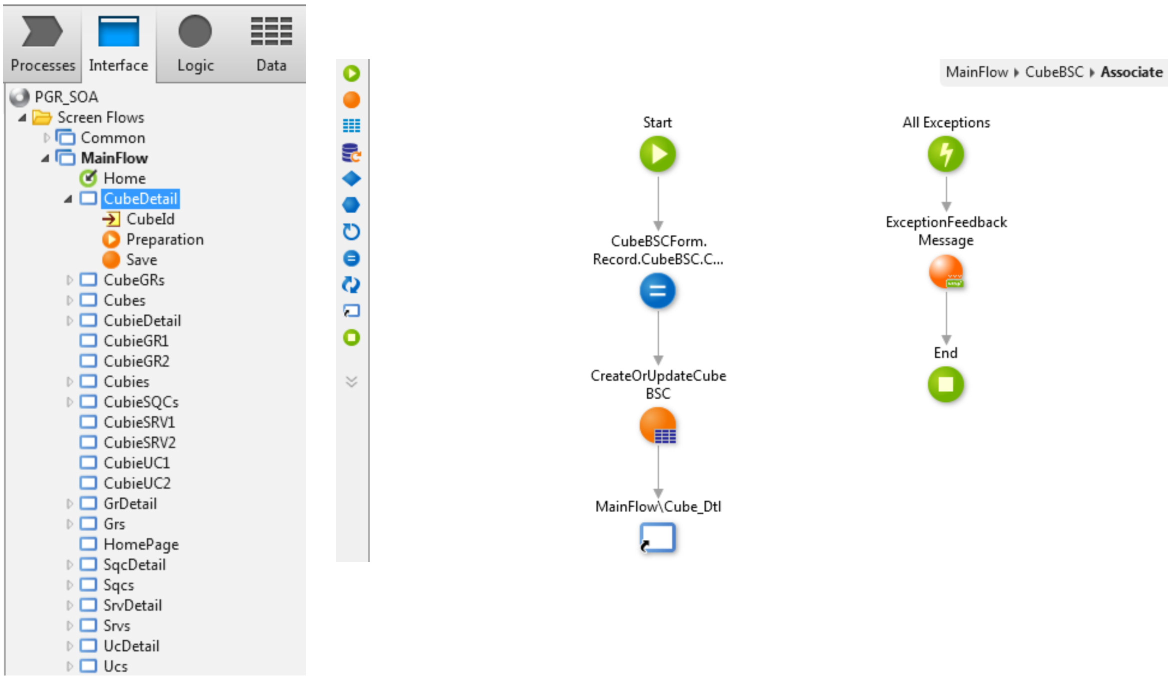


Fig. 6.17: Interface Web Screens, local variables, preparation and events (Associate detail)

Again in the MainFlow, with an initial designed Main screen, the number of screens for this application also more than doubled the ones from the initial PGR application, and kept in line with the ones from the PGR-BSC. Once more, the use of the automatic scaffolding assisted in quickly creating screens to handle chosen entities, just by dragging and dropping each desired entity into the screen canvas space. This generated the typical listing screen for the elements of that entity, with associated filters, attribute sorting and others. As in the previous PGR-BSC, this application demanded many manual configurations to be performed.

For the base entities, by dragging a second turn the same entity in the canvas, a typical detail editing screen for the elements of that entity was also created automatically, with all the links between the two screens and all the needed validations. For the intermediate entities and the more complex relations, as between the Cube and Cubie, many adaptations were also done afterwards on top of the pre-generated screens and on new created screens that were needed (Fig. 6.17). The complete set of generated screens is also shown in Fig. 6.18, where the associated flow between the screens is represented by arrows.

From the Main web screen we can reach the base definition of the goals/rules and SQC items for its dimensions configuration, and visualize the list of existing cubes. Inside a cube detail we can create a new cube, or associate goals/rules, services and use-cases (these last two by creating each one from new), and access the list of existing SQC statements in the already defined cubies.

From the list of cubes we can view each cube in six different configurations, as referred in the previous section, depending on the orientation of the faces for the cube. Independently of the cube face, by clicking on a cubie element, we are able to create/edit a new element for the chosen position and associate the respective SQC statements to it (Fig. 6.18).

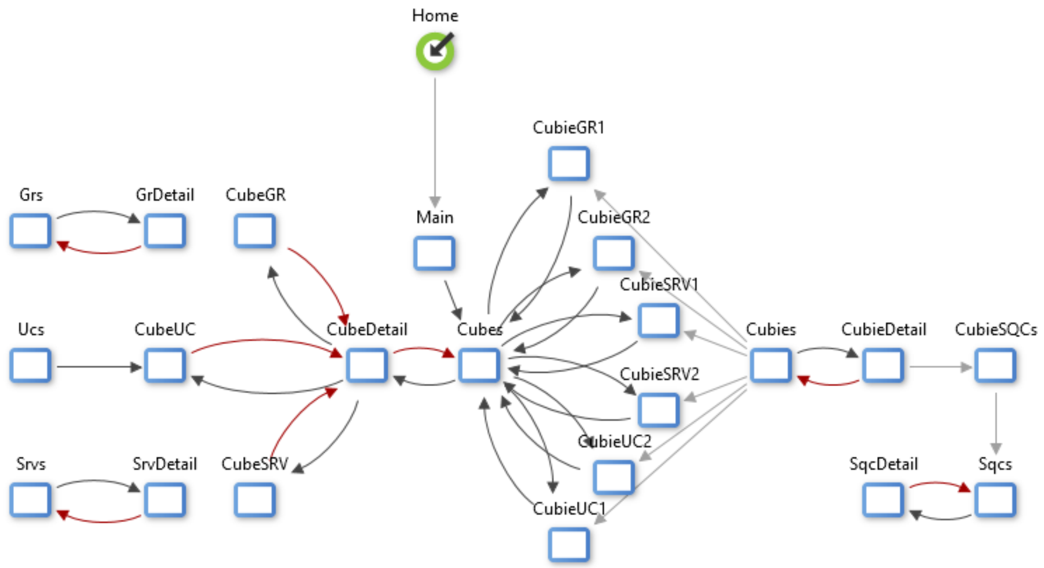


Fig. 6.18: Web Screens flow

Regarding the diverse configurations made in the multiple events of this application, we again highlight the one designed for the Preparation events for the six different web screens to simulate the cube faces rotation (Fig. 6.19). Besides the semi-automatic datasets, some If conditions and For Each cycles, and many variable attributions, with even a handmade SQL expression, were needed. The base configuration is the same for the six different screens, except some minor changes in the datasets entities, and associated adaptations in the attributions and SQL expression. In what relates to the web screens, we highlight the CubeDetail, as it aggregates, besides the name of the cube, all the information related to the configuration of its three dimensions, involving the goals/rules, services and use-cases associations. Its design involves text labels, also recurring to variables, input boxes, buttons, dynamic tables and linked objects (Fig. 6.20). The existence of container elements allow for the correct positioning of several elements in one form.

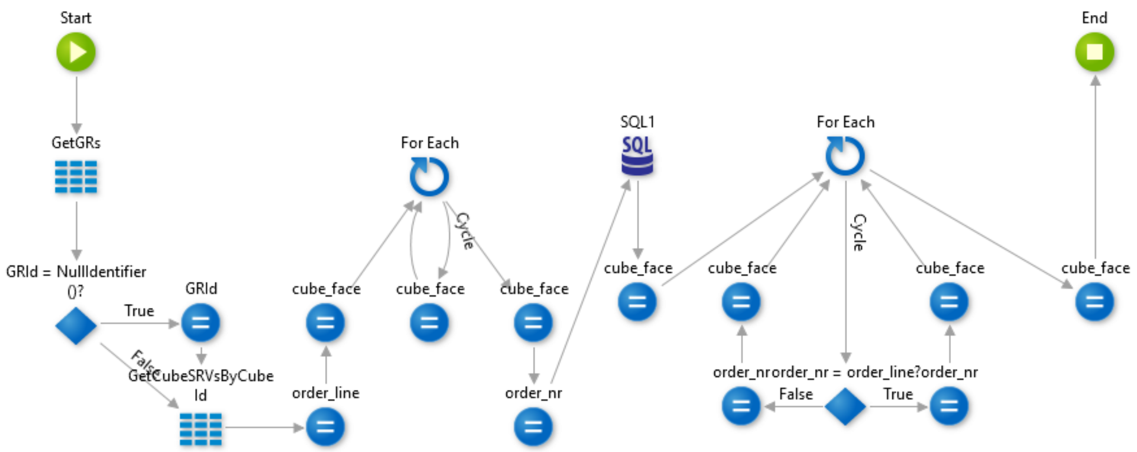


Fig. 6.19: Preparation logic for Web Screen

Once more referring to the encapsulation of the needed business or technical logic in the existing events, besides the initial work set up in the Preparation event to execute prior to its related screen visualization, when any button is pressed there will also be an associated portion of code to execute before continuing to the destination screen. On the other hand, whenever a simple link is followed, it immediately hands out the control to that screen, where the Preparation event is promptly executed.

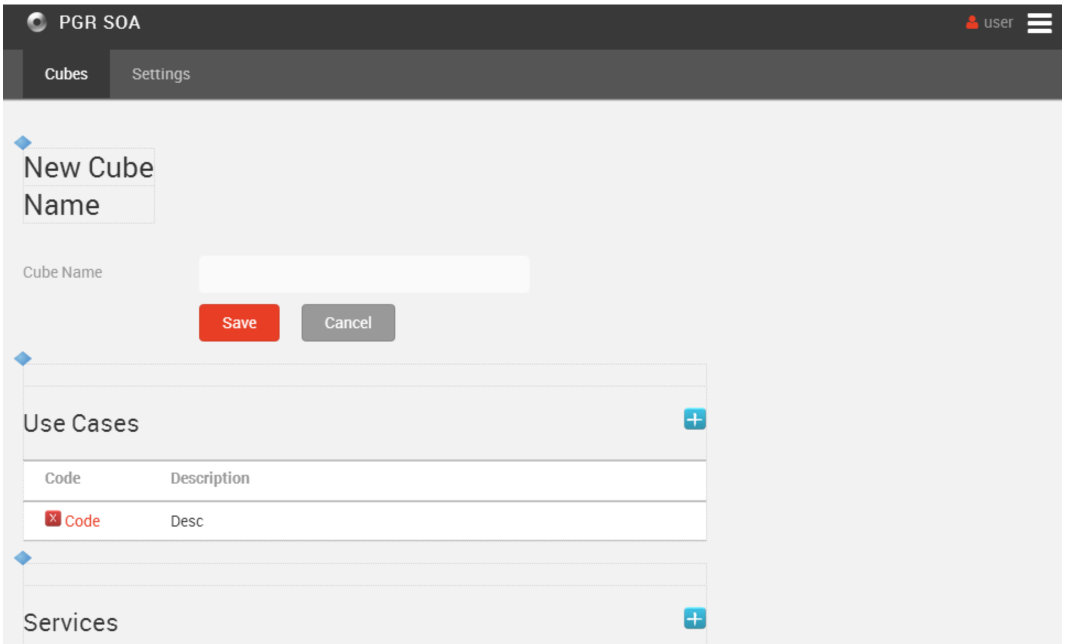


Fig. 6.20: Design mode for Web Screen

6.4.3 Form captions

Lastly, as in the previous applications, after the entities structured, and the screens designed and properly prepared, the application can be generated and deployed to the server. This concluded it is possible to immediately navigate the application and perform the desired operations. In the case of the PGR-SOA Modeler application, the first operation should be to populate the items in the more generic goals/rules and SQC dimensions.

The following task is to create a new cube and associate to it the related items, defining the services and use-cases in the middle of this, in order to compose the different faces of the cube (Fig. 6.21). In the Cube SOA form one can set the name of the cube and associate to it elements from three different dimensions: goals and rules, services and use-cases. From these three, services and use-cases are created in direct dependency from the cube, while the elements from goals and rules have to be previously defined and are merely associated here. Lastly, the complete list of SQC on the cubie elements can be obtained through the SQC Report link.

The main step of defining cubie elements can be performed thereafter, by accessing the cube through one of its six faces (Fig. 6.22) and repeatedly defining new elements, associated to a specific use-case, service and a determined goals/rules dimension. The identification of the SQC and definition of its associated statements in the setting of each cubie can then be performed.

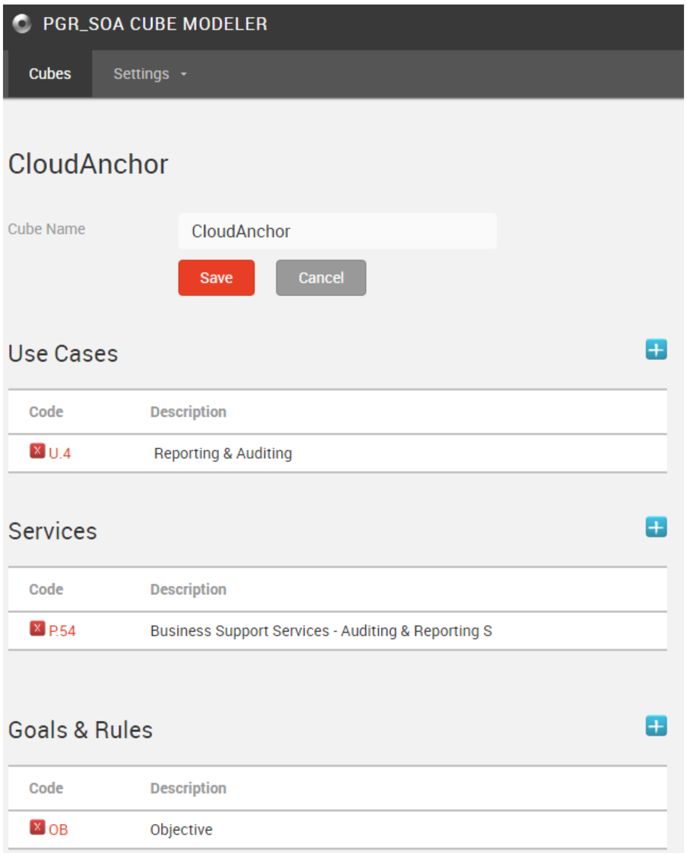


Fig. 6.21: Cube details

So, besides the creation of new cubes, the Cube listing form allows accessing the previously referred details of a cube, by clicking on its name, or to one of the six possible faces of a cube, by clicking on one of the six images on the top-right corner of each cube row. The Cube face form (1 of 6) allows setting, editing or just viewing the cubie elements there available. The example of one of the six faces can be seen in Fig. 6.23, where one dimension is available through a combo-box that, when changed, refreshes the contents of the remaining two-dimension table below.

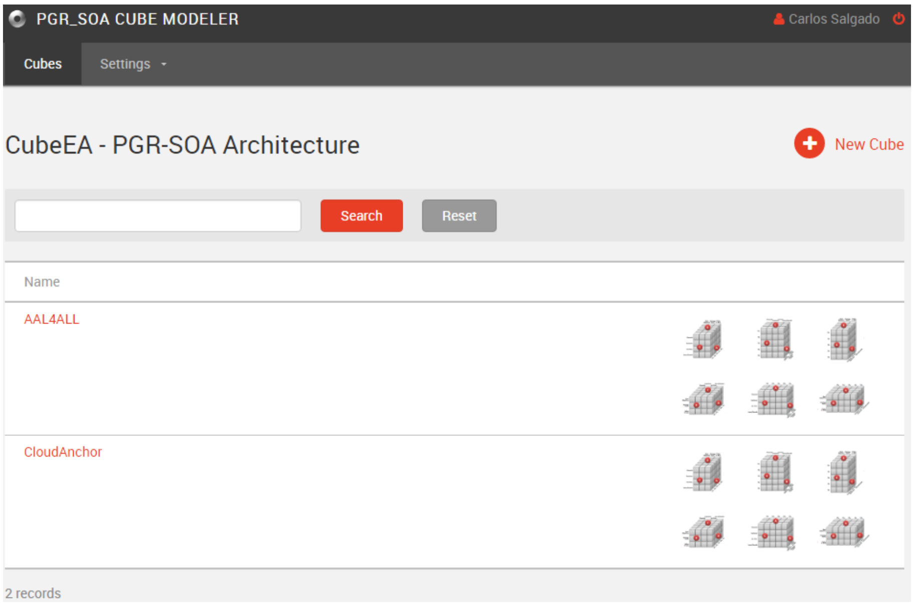


Fig. 6.22: Cube listing

In this particular case the combo-box holds the use-cases items while the goals and rules, and the SRV dimensions constitute, respectively, the vertical and horizontal axis of the table. Also, a simulation for the rotation of the cube can be performed by clicking in one of the six images on the top-right corner of the form, shifting the view throughout the six possible faces.

The cubie statements are the core of the Cube-SOA Modeler application, representing the equivalent to the PGR requirement statements of the previous PGR Modeler or to the PGR-BSC requirement statements of the PGR-BSC Modeler, but now with the added "flavor" of the SRV and the SQC dimensions. The Cubie detail form (Fig 6.24) allows for setting a new cubie element or to edit an already existing one. A cubie element cannot be deleted, as the crossing of the three dimensions is perennial, but it can be left blank if desired.

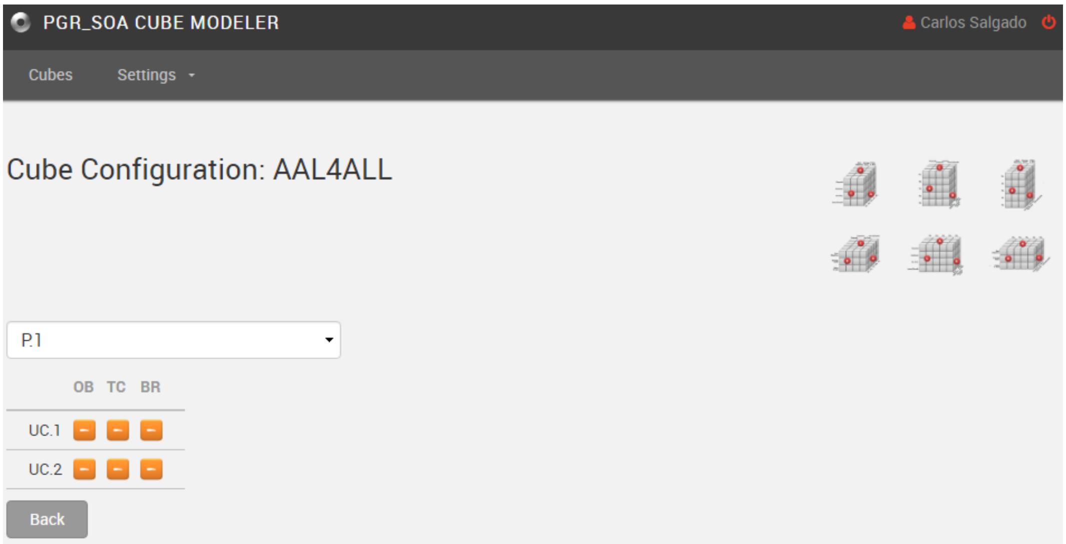


Fig. 6.23: Cube face (1 of 6)

As referred in the previous section, a cubie element is composed by a set of SQC statements, with each statement being adapted to the reality of each SQC type. This operation is the main action to be performed in this Cubie detail form, with the complete set of SQC by use-case, service or goal/rule dimension available from the Cube detail form, also with an option for a complete CISQ report export.

Fig. 6.24: Cubie detail

Appendix B contains examples of the use of this tool regarding the AAL4ALL project. The complete application can be accessed and freely tested in the following web address:

- https://carlossalgado.outsystemscloud.com/PGR_SOA/

6.5 CONCLUSION

SOA is a top concern in research within IS, where the growing use of the SoaML notation can help strengthening the link between business and technological worlds, evermore interconnected and involved in issues of requirements elicitation, process modeling and business strategy. More specifically, the alignment between the business requirements and the quality characteristics of logical architectures is present in a wide range of active topics of research, traversing the requirements engineers and software architects competencies. The combination of the different but connected techniques for elicitation of business requirements, derivation of a logical architecture and generation of quality characteristics, are particularly fit to the business and IS alignment challenges of our research.

Our work integrates all of these topics by initially proposing a way to generate, align and maintain a service-oriented logical architecture for a desired information system. This solution is based on the elicitation of use-cases and their transformation to participants at the end of the 4SRS-SoaML method execution (an evolution from a previous proposal of our research group, the 4SRS method). By detailing and proposing a SPEM-tailored method to generate the logical architecture services quality characteristics from the business requirements, it aligns two different but complementary realities.

Accordingly, as architectures are at the heart of an organizations information system, being closely related to its enterprise, the interrelation of their services with functional and nonfunctional perspectives (business processes, and business goals and rules) allows for an expanded three-dimensional view and control, over the entire strategy and vision of an organization. With the use of modeling features, as well as popular reference models inside the area for this proposal, it allows for further development of the proposed solution and leaves an open door for future connections to other points of interest.

The ongoing and planned work with the PGR, 4SRS-SoaML and CISQ referentials, as well as the support of SPEM, present a promising standpoint from where to further develop, test, evaluate and evolve this research work. Added to the support of a PGR metamodel extension (which connects SoaML participants and CISQ measures with the elicited use-cases, and related business goals and rules for its representation), a method to guide in the alignment of requirements and architectures is fully specified in a tailorable way. This dual solution is then further detailed and analyzed by the proposed PGR-SOA Cube solution, as well as the complemented development of a support tool prototype.

So, in this chapter, we first put forward a visualization to support the connection between each problem-side, leaf-level use-case, with the solution-side, architecture participants. This is built with a strong focus on the requirements elicitation, retaining business information along the way, and aiding in the business/IS alignment and requirements traceability. Therefore, supporting the conscious build of a logical architecture based in participants and super-participants, supported in the SoaML notation, identifying requests and services as communication channels.

Following, the work with the PGR and CISQ framework integration, led to detailing and analyzing the PGR-SOA Cube solution. Complementing the work of the 4SRS-SoaML method, the alignment between PGR-based requirements and SoaML architecture participants closes the gap on the high-level V-Model approach. Similarly to the PGR-BSC Cube, this solution takes advantage of a cube structure and method definition within a SPEM approach, which is adaptable to model variations, contributing to an improved, multi-project, user-friendly, tailorable solution.

Lastly, the development of an assisting support tool to explore and process the different perspectives presents itself as an interesting option for the initial evaluation and testing of our proposal. The use of a rapid-development platform (OutSystems Studio), suited to the initial development in the form of a prototype, allowed shortening paths and deadlines for this solution.

—— This page intentionally left blank ——

CHAPTER 7

CONCLUSION

Summary: This chapter concludes the written part of this document. First, a critical analysis on the initial research question pursued and the proposed goals is performed. Following there is a synthesis of the contributions of this work to the universal body of knowledge, alongside the publications achieved along the research chronogram time span. Finally, the limitations imposed, the lessons learned and the opportunities left open are finalized in the future work topics.

CHAPTER TABLE OF CONTENTS

—— This page intentionally left blank ——

CHAPTER 7

CONCLUSION

"And those who were seen dancing, were thought to be crazy, by those who could not hear the music."

*– Old proverb, **origin unknown***

7.1 CRITICAL ANALYSIS

This thesis drives in the context of the model-based development of information systems and their alignment with business requirements. The motivation arises from the evermore complex challenges presented by this context, both in the creation of new systems, and in the evolution and maintenance of existing systems to adapt to rapidly changing conditions. As information systems are, nowadays, evermore ingrained in the support of business processes within organizations, approaching these challenges is of great importance both to the academic and practice communities.

While existing solutions are not completely able to cope with new ideas, adapt to existing realities or meet increasingly quality demands, the continued evolution and adaptation of methodology-based approaches is essential. Moreover, in our view, the rigorous use of terms, namely by following standards and supporting model-based solutions, is essential to tackle the existing ambiguity and disconnection inside these topics. Also, the same applies to the structuring and consolidation of existing proposals towards agreed-upon solutions, and to clearly stating the differences and added value of new propositions in relation to existing knowledge.

Among the diversity of themes and issues within the described context, the decision to focus on the alignment of business with information systems requirements and architectures comes from the continued need to foster the communication between management and technological realities. This permits to continually identify the organization strategy and customize its processes accordingly. Also, the early steps of the supporting literature review proved essential for understanding what perspectives are involved in aligning information systems with business and its associated topics, and to comprehend in what sense model adoption drives information systems development. All in all, three complementary approaches are proposed throughout this thesis, covering the topics of requirements elicitation, business models and architectures, going from uncovering organizational processes, goals and rules to the derivation of services' architectures.

On the quest to answer the research question of how to develop model-based solutions aligning information systems with business, this thesis focus initially on a prominent V-Model approach, with its associated 4SRS method. This approach has proven valuable in the development of information systems, especially in ill-defined contexts, being able to align problem-solution elements throughout. Concentrating first on the crude, lower-level, transformation between requirements and architecture, where the twin peaks model serves as inspiration, the 4SRS competitors' strengths and weaknesses are analyzed in order to identify the relevant issues where to advance research. Afterwards, the work on extending and broadening the V-Model and 4SRS approaches, detailing their coverage on each of the topics earlier identified, is presented.

Requirements Engineering

Setting the basis for the elicitation of business requirements in order to deliver a sustained, quality-based, development and maintenance of information systems, the choice to use the RUP guidelines and the BMM representation metamodel in the development of an integrated metamodel and method proposal is explored. Answering to the needs of projects for handling quality issues and a wider coverage on the elicitation of business information, while taking into account the influence of recent trends namely from goal-oriented approaches, a proposal for the combined elicitation of functional and nonfunctional requirements is presented.

This proposal for requirements elicitation extends the existing one within the V-Model approach by adding the nonfunctional part (in the form of goals and rules business information), as this previous one is limited to modeling simply their functional side (in the form of use-cases). In a completely modeled and method-driven solution (named PGR) its emphases is now on a twofold rigorous model representation and method specification (grounded in UML, BMM and SPEM), which allows for its customization with other approaches as the recent trend on replacing use-cases with capabilities. Also, by contributing with a tailorable method, this proposal counts with the initial support of a prototype tool for its operationalization and the open possibility to develop other more advanced tools. Nevertheless, it lacks a more definite set of heuristics to guide users in its execution and its broadened coverage of the requirements phase brings an extra sum of human working hours to any project.

Business Models

In order to provide for the generation of business models from the previous referred requirements, while maintaining their mutual alignment and the possibility to trace back and influence the original requirements, the connection between these two realities and two other central references in the area, namely the BSC and the BMC, is considered. Following on the calls from stakeholders for a tighter linkage between the system requirements and the business model of the organization, while pursuing the popularity of the referred models, a proposal for the generation of business models from business requirements within process-level approaches is presented.

This is a completely novel proposal within the V-Model approach, as it did not previously support any modeled representation neither a specification of a method to generate any kind of business model information. The presented solution is inferred from the higher-level functional and nonfunctional elicited requirements, and complemented by an intersection with the business-side perspectives of the BSC, which gives origin to a more business-like representation delivered in a strategy-oriented view. Added to a tailorable method specification and the corresponding support prototype tool, an optional mapping between the BSC and BMC models is also integrated within the presented proposal. Due to the complexity involved in handling the diverse perspectives, a three-dimensional approach on this solution is proposed for better user perception, and added flexibility in different visualization and processing techniques (the PGR-BSC Cube).

Enterprise Architectures

By arranging for the derivation of architectures from the business requirements, while attaining alignment and traceability for their mutual transformation and adaptation, the circle around these fundamental elements of the problem-solution paradigm is closed. Contrasting with the business model proposal the initial pieces for the architecture derivation are the lower-level requirements, the ones closer to the system details. Regarding the architectural side, the choice is directed to service-oriented architectures (here represented by SoaML) which stand nowadays as a prominent trend in information systems. Furthermore, the need to associate the quality issues (initially elicited in the business domain) with the ones in the information system implementation scenarios, led to recurring to the use of the CISQ-SQC measures.

The present proposal evolves the conventional 4SRS method, adding elements to the model support around its deliverables and adapting its functional steps to them, and includes a second method specification for the post-alignment of the initial nonfunctional elicited requirements with the derived architecture participants' quality characteristics. On one hand, the new 4SRS-SoaML mimics the steps of the original method but takes advantage of the characteristics of services to deliver a more consolidated, business-oriented, solution. On the other hand, the all new post-alignment method is driven by the CISQ-SQC measures, which allows delivering the quality information elicited in the requirements closer to the architectural implementation levels. Within the customable method specification proposed and its corresponding support prototype tool, the mapping between these two realities is made in a more straightforward and dynamic way. Nevertheless, again due to the complexity involved in handling the diverse perspectives within, a three-dimensional approach on this solution is proposed (the PGR-SOA Cube).

Final considerations

Altogether, guided by the challenge of aligning business requirements with information systems and grounded in the established V-Model and 4SRS methods, this work advances the existing solutions in three directly related topics, namely requirements engineering, business models and enterprise architectures. Nevertheless, although the proposed PGR requirements elicitation, PGR-BSC Cube business model approach, and 4SRS-SoaML architecture derivation method and its associated PGR-SOA Cube approach are all interconnected, they are presented and developed individually, so they still do not constitute a totally integrated solution yet.

On the other hand, as modeling traverses all the proposed approaches and due to the choice of following OMG model-based solutions, the future evolution to a completely integrated approach stands as the natural progression. Accordingly, the use of a transversal specification as SPEM in the definition of the proposed methods already allowed for the fast development of supporting tools in the form of prototypes, using the OutSystems Studio development platform. Moreover, it also permits a high-level of flexibility in the customization of the methods for use in diverse projects and by different user profiles, and also for the future development of more advanced and wider integrated support tools.

The widespread use of OMG specifications, under the ever-present general-purpose UML, and the more specific business-side BMM and architecture-side SoaML and CISQ standards, allow to, simultaneously, build on existing knowledge and contribute with open solutions to the academic and practice communities. It also eases communication with peers and strengthens evaluation due to their solid grounding. Accordingly, the same applies to the choice for, whenever possible, developing the proposed solutions around established reference models as RUP, BSC, BMC and SOA. Even so, care should be taken not to constrain the choices made available to users when facing problems that demand solutions outside these standards and reference models scope.

The entire research process, including the main question, its associated goals and the corresponding activities follow upon the DSR guidelines, covering its three cycles: rigor by grounding on existing knowledge and adding to the knowledge base; relevance by supporting on environment requirements and field tests with activities in academic/industrial projects; and design by iterating and refining the build/design of artifacts, continually evaluating the obtained solutions. The choice for the adopted research method, mixing design science and action research, allowed connecting the research question and goals with the planned activities in the research plan chronogram. From the initial state of the art on each topic and its continued research, through the persistent development of the needed artifacts by repeatedly evaluating them and communicating results, all resulted in valuable contributes to this thesis.

7.2 CONTRIBUTIONS

As any doctoral thesis aims to contribute to the universal body of knowledge, its contributions should be well stated and revised. Accordingly, we revisit our contributions through the view for excellence proposed in the form of the seven DSR guidelines which, as previously stated in Chapter 1, are useful for understanding, executing and evaluating design science. Others could do a better job in assessing our contributions, as no one can act totally oblivious to its own work, nevertheless, we propose to synthetize the work performed in this thesis for each one of the guidelines:

1. Problem Relevance – Must produce a viable artifact.

   Throughout the work for this thesis we have produced purposeful artifacts, created to address important organizational problems, such as crossing functional and nonfunctional requirements in the elicitation of a business domain, in connecting these with a respective business model specification and in aligning the former with a derived solution service-oriented architecture, ones unveiled from the literature reviews we performed and others answering to the needs of the projects where we collaborated. Moreover, these have been effectively described in the reports delivered to projects, with a synthesis of these later presented in appendices A and B. Also, besides enabling their implementation and application in appropriate domains, namely through development of associated prototypes to support their use in projects or demonstrations, we include not only their instantiations but also the constructs, models and methods applied in their development and use, also complemented with a synthesis of their application details in appendices A and B.

2. Research Rigor – Produces technology-based solutions to relevant business problems.

   Following on the previous guideline, this research work also fulfills the objectives of: acquiring knowledge and understanding, namely through the extensive background review of associated literature; enabling the development and implementation of technology-based solutions, acting mainly on the software engineering side of projects; and motivating on unsolved and important business problems. Moreover, the results of this research work are relevant to diverse communities on the different topics identified, both in diverse business and technological ones. Also, it addresses the problems faced and the opportunities afforded by the interaction of people, organizations and IT, both in academy and industry, delivering effective artifacts that count with constructs by which to think about, models by which to represent and explore, methods by which to analyze or optimize, and instantiations that demonstrate how to affect them.

3. Design as a Search Process – Evaluation that demonstrates of utility, quality, and efficacy.

Regarding evaluation, this thesis includes the integration of the produced artifacts within the technical infrastructure of business environments. It develops solutions through execution and delivers associate support tools, while using methodologies available in the knowledge base, namely observational (case studies) and descriptive (informed argument and scenarios), with a synthesis of these presented in appendices A and B. During this research work a continuous evaluation of the design artifacts was performed, so providing essential feedback to their construction phases, especially as to the quality of the solutions under development. The design artifacts were continuously evolved in a development cycle from basic to complex solutions, until they satisfied the requirements and constraints of the problems they were meant to solve.

4. Design as an Artifact – Research contribution of the design artifact, foundations, or methodologies.

The three integrated approaches proposed, namely the PGR, PGR-BSC and PGR-SOA, all provide contributions in the areas of the design artifact and the design construction knowledge. The respective metamodel and method artifacts are based in reference models, standards and methodologies collected from the knowledge base. These bring both novelty in the way they are adapted and evolved into innovative solutions, generality due to their open model-based development basis and significance in the instantiated prototypes applied to live business and technology environments. So, through the IS research cycles performed they extend the existing knowledge base and are applied in live organizational environments, remaining faithful to the representations they extend from the knowledge base and implementable through the prototype support tools exposed description. Accordingly, they contribute to the business environment and answer previously unsolved problems, as reported in the projects synthetized in appendices A and B.

5. Design Evaluation – Rigor in construction and evaluation method.

Rigorous methods were applied in both the construction and evaluation of the designed artifacts, methods derived from the effective use of the knowledge base-theoretical foundations and research methodologies reviewed in the IS theory, as referred in Guideline 2. Also, we believe to have constructed in this research work both rigorous and relevant models, methods and instantiations. The applicability and generalizability of the artifacts were exercised within appropriate environments, respectively by the execution of projects counting on the support of prototypes and by following an open, standard definition, on the models and methods. The artifacts' evaluation aimed to determine how well each artifact worked, either alone or integrated in a solution, as they were initially supported in standard specifications, then developed in an academic environment and finally deployed in meaningful projects, as reported in the projects synthetized in appendices A and B.

6. Research Contributions – A problem-situated means-ends search for an effective artifact.

The research work for this thesis consisted in an evolving search process to discover effective solution to a problem, directly associated to the posed research question further divided in three related goals. By utilizing the available means to reach the desired ends this process involved creative and innovative proposals, with progress made iteratively as the scope of the design problem was expanded in the search for evermore satisfactory solutions. This search process was conducted iteratively, either identifying deficiencies in the models and methods specifications other pinpointing improvements for the instantiated prototype software systems. It relied on the use of heuristics to create worthy design solutions, always acting with the purpose to address the pending challenges, in a constantly evolving development process of the proposed approaches in breadth and complexity.

7. Research Communication – Communication to both technical and managerial audiences.

All contributions of this thesis were widely communicated, both to technology-oriented as well as to management-oriented audiences. The primary channels for disseminating this work were the presentation of papers in conferences and the joint writing of project reports. In both of these, interactions with fellow researchers and diverse stakeholders from different areas allowed to discuss in detail several issues regarding the proposed approaches. In terms of the contributions, the use of 'technical' languages as SPEM, BMM and SoaML serves the technology-oriented audiences, detailing the artifacts to be implemented and used within an appropriate organizational context. Likewise, the use of organization reference models as BSC and BMC serves the management-oriented audiences, helping to determine the organizational resources needed in order to build and use the artifacts within their specific organizational context.

Overall, the research process followed in this thesis went through the three traditional cycles of the DSR process. Initially, the relevance cycle was informally used to identify opportunities and problems within our own work environment and associated interests. Following, the intention to ground our research on scientific foundations of the knowledge base led us to perform the rigor cycle with an early literature review. Then, the design cycle involved building and evaluating each design artifact, continuously refining their design, while iterating through the other cycles for validation and grounding, until satisfactory results were achieved.

Three sets of artifacts are proposed in this thesis, one for each researched topic. First, in chapter 4, the PGR metamodel and its associated method, developed for eliciting business requirements in the form of functional and nonfunctional elements, as well as its support tool, the PGR Modeler, are presented. Then in chapter 5, the PGR-BSC Cube metamodel and its associated method, developed for generating a business model from business requirements, as well as its support tool, the PGR-BSC Modeler, are described. Finally, in chapter 6, the 4SRS-SoaML method and the PGR-SOA Cube approach are presented. The 4SRS-SoaML stands as an evolution of the traditional 4SRS method for the transformation between requirements and architectures, relating use-cases and services participants. Accordingly, the PGR-SOA approach contributes with a metamodel and its associated method, as well as its support tool, the PGR-SOA Modeler, to close the gap around functional and nonfunctional elements, relating these last to the services quality characteristics.

Accomplishments

Referring back to the research chronogram, after the initial work that ended in July 2012 with the presentation of the research plan for this thesis there was time to advance with the preparation for the initial thesis proposal presentation. Besides the initial literature review and base thesis writing itself, our research is distributed into three main segments, each of them relating to an answer to a subsidiary research goal, which originate the respective artifacts to be produced, namely in Requirements Engineering (RE), Business Model (BM) and Enterprise Architecture (EA) (Fig 7.1).

The initial thesis work resulted in the publication of two papers, being the first one derived from the initial literature review for this proposal, with a focus on the central role of requirements engineering in the alignment topics. The second counts with the first steps on the basis elicitation of requirements to integrate with the business model artifact representation, its relation with strategy and goal information, and the use of standard specifications within a new proposed method, already integrated in the RE segment of our research.

The first published paper was presented at the ACM 28th Symposium On Applied Computing, in March 2013, at Coimbra, Portugal [Salgado, Machado, & Suzana, 2013], while the second was a paper presentation at the 22nd International Conference on Information Systems Development, in September 2013, at Seville, Spain (Springer) [Salgado, Machado, & Maciel, 2014b]. In between these two presentations, the public defense of the initial proposal for this thesis took place in July 2013, at the Universidade do Minho, Guimarães, Portugal.

These first publications focused on the requirements engineering perspective of modeling the alignment between business and IS/IT, and on the continued theoretical development of the requirements representation and its associated elicited guidelines (RE in Fig. 7.1). Some other papers were planned on this topic to publish further innovations, but the lack of projects where to apply them, the tight timeframe available and our quest for quality publishable-associated venues, made it close to impossible.

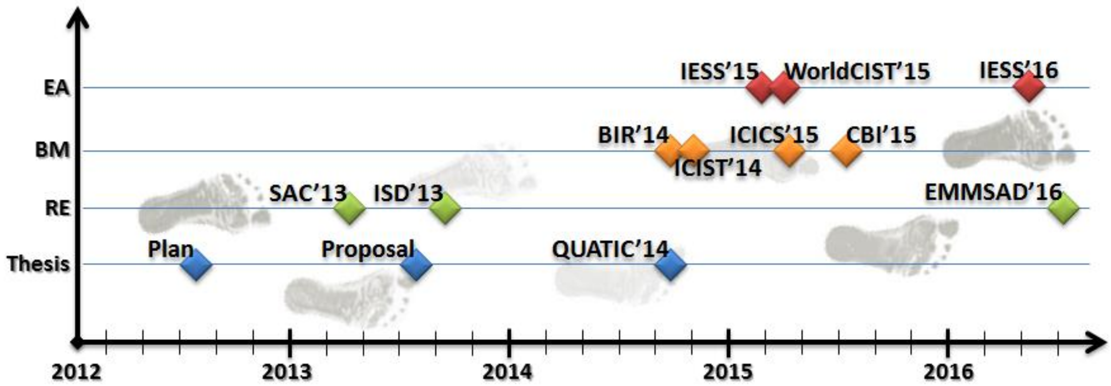


Fig. 7.1: Research chronogram

Early on 2014, the chance to collaborate on a new project was proposed, within the AC.pt project as presented in Chapter 5, where our intervention was directed at the proposal of a business model for the information system to be developed. The successful collaboration inside this project gave rise to a series of papers within the business model topic, being the first at the 13th International Conference on Perspectives in Business Informatics Research, in September 2014, at Lund, Sweden (Springer) [Salgado, Teixeira, Machado, & Maciel, 2014a]. A second paper introducing some more formality to the previous research, with the use of SPEM, was presented at the 20th International Conference on Information and Software Technologies, in October 2014, at Druskininkai, Lithuania (Springer) [Salgado, Teixeira, Machado, & Maciel, 2014b].

The research work on the business model topic continued in 2015, with further innovations on business models being presented at two conferences, one at the 6th International Conference on Information and Communication Systems, in April 2015, at Amman, Jordan (IEEE) [Salgado, Machado, & Maciel, 2015a], and the other at the 17th IEEE Conference on Business Informatics, in July 2015, at Lisbon, Portugal (IEEE) [Salgado, Machado, & Maciel, 2015c]. An additional paper, in the form of a tool demonstration, was accepted for presentation at the 23rd IEEE International Requirements Engineering Conference, in August 2015, at Ottawa, Canada (IEEE), but in the end we had to cancel our participation due to logistic problems.

This second segment represents our research work on a method for the generation of a business model (BM in Fig. 7.1), supported by the elicitation of requirements and their mapping to a business model reference model. Alongside it, there is the proposal for a tailorable specification of this method within a model-based approach and its generalization in the form of a cube structure, allowing to easily move back and forth between the business model and the base requirements.

During the course of 2014 another opportunity to collaborate on a project was positively answered, within the AAL4ALL project as presented in Chapter 6. As this project was already in an advanced stage, our contribution was directed to the transformation between requirements and architecture. Again, this cooperation led to a series of paper publications, one at the 6th International Conference on Exploring Service Science, in February 2015, at Oporto, Portugal (Springer) [Salgado, Teixeira, Santos, Machado, & Maciel, 2015], and another at the 3rd World Conference on Information Systems and Technologies, in April 2015, at Ponta Delgada, Azores, Portugal (Springer) [Salgado, Machado, & Maciel, 2015b]. In the following year, 2016, a third paper in this research segment was accepted for presentation at the 7th International Conference on Exploring Service Science, held in May 2015, at Bucharest, Romania (Springer) [Salgado, Machado, & Maciel, 2016a].

This third perspective of our work involves a proposal to adapt a method from within our research group (EA in Fig. 7.3), which has proven successful in generating a proper candidate logical architecture for an information system in ill-defined contexts to a service-oriented architecture approach, in order to achieve greater business integration, flexibility and agility. This allows for greater integration of the elicited requirements, namely to its alignment with the generated architecture quality characteristics, while the use of model-based notations and, again, a cube structure that permit the generalization and tailorable specification of the method.

Besides these contributions in the three research topics, during 2014, roughly through half of this research work, we had the possibility to further validate our research plan with the participation in the Doctoral Consortium of Portuguese Software Engineering Students, collocated at the 9th International Conference on the Quality of Information and Communications Technology, held in September 2014, at Guimarães, Portugal (IEEE) [Salgado, Machado, & Maciel, 2014a]. In the final steps of this work, near the conclusion of the thesis writing, a paper on the analysis and comparison of approaches for the transformation between requirements and architectures was successfully published on the 21st International Conference on Exploring Modelling Methods for Systems Analysis and Design, in June 2016, at Ljubljana, Slovenia (Springer) [Salgado, Machado, & Maciel, 2016b].

7.3 FUTURE WORK

The thematic of alignment between business and information systems remains a popular topic, being a top concern of people in organizations, but with no mainstream consensual solutions.

Despite the many efforts undertook in the field, existing approaches to tackle its issues are still very fragmented and dispersed. We believe that the results derived from the research question and goals of this thesis contribute to expand the common knowledge base and adopt innovative methods within organizational environments. Nevertheless, we never had the intention to be able to definitively solve all the problems stated, as so much work is left to do.

In what concerns the development of information systems, the continuous efforts of the research and practice communities need to keep evolving in order to address existing and upcoming challenges. Existing trends increasingly approach managerial and IS practices, levelling the distance between decision and implementation stages, calling for a deeper involvement of stakeholders. Also, the focus in deploying customizable solutions permits for alignment choices that are adaptable to rapid changes in the organizational environments and innovations in information systems.

Despite the importance of other related topics, we still emphasize the continued relevance of advancing research in requirements engineering (RE), business models (BM) and enterprise architectures (EA) issues, as followed in this thesis. In all these topics, the importance of alongside the proposal of well-grounded innovative solutions, to continually carry out case studies and controlled experiments, is crucial in advancing and consolidating the research findings and existing practices in the community.

RE is an established field, with traditional elicitation techniques remaining a firm choice for researchers and practitioners. Even so, new efforts in a diversity of areas still emerge, such as; the creative elicitation of requirements, more sophisticated modeling of domain knowledge and model-checking tools; the growth in natural language (NL) tools and social collaboration support, with better prospects for the so called green-field in contrast with brown-field domains; and the elicitation of unknown requirements [Sutcliffe & Sawyer, 2013].

Regarding contemporary trends, the handling of nonfunctional requirements is already consensual among the RE communities, while goal-oriented RE (GORE) counts with a wide number of followers. Other interesting approaches are the recognized goal question metric (GQM) [Solingen, Basili, Caldiera, & Rombach, 2002], alongside a panoply of derivative work, and the recent studies on enterprise capabilities [Stirna, Grabis, Henkel, & Zdravkovic, 2012]. In what concerns our PGR proposal, it would be interesting to face future customizations by using capabilities instead of use-cases or GQM reference items in place of the goals-rules BMM details.

Oppositely, BM has had advances and drawbacks, with considerably important but yet insufficiently researched trends. Specifically regarding the reference models with which we work more closely, the BMC has been losing momentum while the BSC has revived in recent years. Other topics have been tested and presented, namely the design and management of sustainable business models, interrelating and connecting the business models perspective to the established concepts of corporate sustainability or sustainable innovation [Boons & Lüdeke-Freund, 2013].

Accordingly, the question on how companies change and develop their business models explores associated issues to achieve sustained value creation [Achtenhagen, Melin, & Naldi, 2013]. Moreover, the strategic management researchers' study of the design of business models and other similar strategic notions can profit from deeper IS contributions, due to its more conceptual and design-oriented research [Osterwalder & Pigneur, 2013]. As for future work involving our PGR-BSC approach, it can also be easily customized and adapted to use other reference models, or further integrated with the PGR-SOA approach.

EA is a quite clustered topic, with no precise definition of what constitutes EA research. Different schools of thought exist, building on the context and contents of one another, but with fundamental differences in their assumptions and values. While some reductionist enterprise architecture practices lack acceptance and are perceived as organizationally insensitive, other holistic ways of thinking allow to consistently deliver adaptation or innovation [Lapalme, 2012]. Most promising and challenging themes, from a practical point of view, involve a more balanced examination of certain architectural layers, method elements, EA management tasks, and lifecycle phases.

Also, as practitioners acknowledge that service orientation has become a prevalent concept, they embrace the cloud as a natural extension of the enterprise architecture [Simon, Fischbach, & Schoder, 2013]. Also, service-oriented EA in the cloud are quickly emerging and shaping trends, which leads to an integration model for their design, and the systematic development of architecture artifacts of service-oriented cloud-based enterprise systems for big data applications [Zimmermann *et al.*, 2013]. Regarding our PGR-SOA approach, further work on the questions of quality and architectural patterns, as well as on the 4SRS method and its derivatives, is inevitable.

References

Abran, A., & Buglione, L. (2003). A Multidimensional Performance Model for Consolidating Balanced Scorecards. *Advances in Engineering Software*, *34*(6), 339--349.

Achtenhagen, L., Melin, L., & Naldi, L. (2013). Dynamics of Business Models – Strategizing , Critical Capabilities and Activities for Sustained Value Creation. *Long Range Planning*, *46*(6), 427--442.

Aguilar-Savén, R. S. (2004). Business process modelling: Review and framework. *International Journal of Production Economics*, *90*(2), 129--149.

Al-Debei, M. M., & Avison, D. (2010). Developing a unified framework of the business model concept. *European Journal of Information Systems*, *19*(3), 359--376.

Alter, S. (2008). Defining information systems as work systems: implications for the IS field. *European Journal of Information Systems*, *17*(5), 448--469.

Ambler, S., Nalbone, J., & Vizdos, M. (2005). *The Enterprise Unified Process: Extending the Rational Unified Process*. Prentice Hall.

Ameller, D., Ayala, C., Cabot, J., & Franch, X. (2013). Non-functional Requirements in Architectural Decision Making. *IEEE Software*, *30*(2), 61--67.

Amyot, D., Ghanavati, S., Horkoff, J., Mussbacher, G., Peyton, L., & Yu, E. (2010). Evaluating goal models within the goal-oriented requirement language. *International Journal of Intelligent Systems*, *25*(8), 841--877.

Anaya, V., Berio, G., Harzallah, M., Heymans, P., Matulevičius, R., Opdahl, A. L., … Verdecho, M. J. (2010). The Unified Enterprise Modelling Language—Overview and further work. *Computers in Industry*, *61*(2), 99--111.

Andersson, B., Bergholtz, M., Edirisuriya, A., Ilayperuma, T., Johannesson, P., Gordijn, J., … Weigand, H. (2006). Towards a reference ontology for business models. In *Conceptual Modeling-ER 2006* (pp. 482--496). Springer.

Armour, F. J., Kaisler, S. H., & Liu, S. Y. (1999). A big-picture look at enterprise architectures. *IT Professional*, *1*(1), 35--42.

Asosheh, A., Nalchigar, S., & Jamporazmey, M. (2010). Information technology project evaluation: An integrated data envelopment analysis and balanced scorecard approach. *Expert Systems with Applications*, *37*(8), 5931--5938.

Atkinson, H. (2006a). Strategy implementation: a role for the balanced scorecard? *Management Decision*, *44*(10), 1441--1460.

Atkinson, H. (2006b). Strategy implementation: a role for the balanced scorecard? *Management Decision*, *44*(10), 1441–1460.

Avison, D., Jones, J., Powell, P., & Wilson, D. (2004). Using and validating the strategic alignment model. *The Journal of Strategic Information Systems*, *13*(3), 223--246.

Banker, R. D., Davis, G. B., & Slaughter, S. A. (1998). Software Development Practices, Software Complexity and Software Maintenance: A Field Study. *Management Science*, *44*(4), 433--450.

Banker, R. D., & Kauffman, R. J. (2004). The Evolution of Research on Information Systems: A Fiftieth Year Survey of the Literature in Management Science. *Management Science*, *50*(3), 281--298.

Berkem, B. (2008). From the business motivation model (BMM) to Service Oriented Architecture (SOA). *Journal of Object Technology*, *7*(8), 57--70.

Boons, F., & Lüdeke-Freund, F. (2013). Business models for sustainable innovation: State-of-the-art and steps towards a research agenda. *Journal of Cleaner Production*, *45*(4), 9--19.

Börger, E. (2012). Approaches to modeling business processes: a critical analysis of BPMN, workflow patterns and YAWL. *Software and Systems Modeling*, *11*(3), 305--318.

Braun, C., & Winter, R. (2005). A Comprehensive Enterprise Architecture Metamodel and Its Implementation Using a Metamodeling Platform. In *Proceedings of the Workshop on Enterprise Modelling and Information Systems Architectures* (pp. 64--79). GI.

Breivold, H. P., Crnkovic, I., & Larsson, M. (2012). A systematic review of software architecture evolution research. *Information and Software Technology*, *54*(1), 16--40.

Brinkkemper, S. (1996). Method engineering: Engineering of information systems development methods and tools. *Information and Software Technology*, *38*(4), 275--280.

Brown, C. V, & Magill, S. L. (1994). Alignment of the IS functions with the enterprise: toward a model of antecedents. *MIS Quarterly*, *18*(4), 371--403.

Buglione, L., & Abran, A. (2000). Balanced Scorecards and GQM: What are the Differences? In *Proceedings of FESMA/AEMES Software Measurement Conference* (pp. 1--10).

Buglione, L., & Abran, A. (2002). QEST nD: n-dimensional extension and generalisation of a software performance measurement model. *Advances in Engineering Software*, *33*(1), 1--7.

Burkhart, T., Krumeich, J., Werth, D., & Loos, P. (2011). Analyzing the Business Model Concept-A Comprehensive Classification of Literature. In *Proceedings of the 32nd International Conference on Information Systems* (pp. 1--19). AIS.

Cabot, J., Pau, R., & Raventós, R. (2010). From UML/OCL to SBVR specifications: A challenging transformation. *Information Systems*, *35*(4), 417--440.

Cardoso, E. C. S., Almeida, J. P. A., & Guizzardi, G. (2009). Requirements engineering based on business process models: A case study. In *Proceedings of the 13th Enterprise Distributed Object Computing Conference Workshops* (pp. 320--327). IEEE.

Casadesus-Masanell, R., & Ricart, J. E. (2010). From Strategy to Business Models and to Tactics. *Long Range Planning*, *43*(2), 195--215.

Castro, J., Lucena, M., Silva, C., Alencar, F., Santos, E., & Pimentel, J. (2012). Changing attitudes towards the generation of architectural models. *Journal of Systems and Software*, *85*(3), 463--479.

Chan, Y. E., & Reich, B. H. (2007). IT alignment: what have we learned? *Journal of Information Technology*, *22*(4), 297--315.

Cheng, B., & Atlee, J. (2007). Research Directions in Requirements Engineering. In *Proceedings of the Conference on The Future of Software Engineering* (pp. 285--303). ACM.

Chung, L., Cesar, J., & Leite, P. (2009). On Non-functional Requirements in Software Engineering. In *Conceptual modeling: Foundations and applications* (pp. 363--379). Springer.

Cleland-Huang, J., Hanmer, R. S., Supakkul, S., & Mirakhorli, M. (2013). The Twin Peaks of Requirements and Architecture. *IEEE Software*, *30*(2), 24--29.

Clements, P. C. (2000). *Active Reviews for Intermediate Designs*.

Conboy, K. (2009). Agility from first principles: Reconstructing the concept of agility in information systems development. *Information Systems Research*, *20*(3), 329--354.

Cooper, K., Abraham, S. P., Unnithan, R. S., Chung, L., & Courtney, S. (2006). Integrating visual goal models into the Rational Unified Process. *Journal of Visual Languages & Computing*, *17*(6), 551--583.

Croll, P. R. (2008). Quality attributes - Architecting systems to meet customer expectations. In *Proceedings of the 2nd Annual Systems Conference* (pp. 1--8). IEEE.

Daneva, M., Buglione, L., & Herrmann, A. (2013). Software architects’ experiences of quality requirements: what we know and what we do not know? In *Requirements Engineering: Foundation for Software Quality* (pp. 1--17). Springer.

Dardenne, A., Van Lamsweerde, A., & Fickas, S. (1993). Goal-directed requirements acquisition. *Science of Computer Programming*, *20*(1), 3--50.

DaSilva, C. M., & Trkman, P. (2014). Business Model: What it is and What it is Not. *Long Range Planning*, *47*(6), 379--389.

Davies, I., Green, P., Rosemann, M., Indulska, M., & Gallo, S. (2006). How do practitioners use conceptual modeling in practice? *Data & Knowledge Engineering*, *58*(3), 358--380.

Davis, G. (2000). Information Systems Conceptual Foundations: Looking Backward and Forward. In *Organizational and Social Perspectives on Information Technology* (pp. 61--82).

de Boer, R. C., & van Vliet, H. (2009). On the similarity between requirements and architecture. *Journal of Systems and Software*, *82*(3), 544--550.

Delgado, A., Guzmán, I. G. De, & Piattini, M. (2010). From BPMN business process models to SoaML service models: a transformation-driven approach. In *Proceedings of the 2nd International Conference on Software Technology and Engineering* (pp. 314--319). IEEE.

Delgado, A., & Ruiz, F. (2012). Model transformations for Business-IT alignment: from collaborative business process to SoaML service model. In *Proceedings of the 27th Annual ACM Symposium on Applied Computing* (pp. 1720--1722). ACM.

Delgado, A., Ruiz, F., Guzmán, I. de, & Piattini, M. (2009). Towards a Service-oriented and Model-driven Framework with Business Processes as First-class Citizens. In *Proceedings of the 2nd International Conference on Business Process and Services Computing* (pp. 19--31). GI.

Demirkan, H., Kauffman, R. J., Vayghan, J. a., Fill, H.-G., Karagiannis, D., & Maglio, P. P. (2008). Service-oriented technology and management: Perspectives on research and practice for the coming decade. *Electronic Commerce Research and Applications*, *7*(4), 356--376.

Dingsøyr, T., Nerur, S., Balijepally, V., & Moe, N. B. (2012). A decade of agile methodologies: Towards explaining agile software development. *Journal of Systems and Software*, *85*(6), 1213--1221.

El-Attar, M., & Miller, J. (2012). Constructing high quality use case models: a systematic review of current practices. *Requirements Engineering*, *17*(3), 187--201.

Elvesaeter, B., & Panfilenko, D. (2010). Aligning business and IT models in service-oriented architectures using BPMN and SoaML. In *Proceedings of the 1st International Workshop on Model-Driven Interoperability* (pp. 61--68). ACM.

Engelsman, W., Quartel, D., Jonkers, H., & van Sinderen, M. (2011). Extending enterprise architecture modelling with business goals and requirements. *Enterprise Information Systems*, *5*(1), 9--36.

Engelsman, W., & Wieringa, R. J. (2012). Goal-Oriented Requirements Engineering and Enterprise Architecture: Two Case Studies and Some Lessons Learned. In *Requirements Engineering: Foundation for Software Quality* (pp. 306--320). Springer.

Fatolahi, A., & Shams, F. (2006). An investigation into applying UML to the Zachman framework. *Information Systems Frontiers*, *8*(2), 133--143.

Ferreira, N., Santos, N., Machado, R., & Gašević, D. (2013). Aligning Domain-related Models for Creating Context for Software Product Design. In *Increasing Value in Software and Systems Development Software Quality* (pp. 168--190). Springer.

Ferreira, N., Santos, N., Machado, R. J., Fernandes, J. E., & Gasevic, D. (2014). A V-Model Approach for Business Process Requirements Elicitation in Cloud Design. In *Advanced Web Services* (pp. 551--578). Springer.

Fitzgerald, G., & Avison, D. E. (2003). Where now for Development Methodologies? *Communications of the ACM*, *46*(1), 79--82.

France, R., & Rumpe, B. (2007). Model-driven Development of Complex Software: A Research Roadmap. In *Proceedings of the Conference on Future of Software Engineering* (pp. 37--54). IEEE.

Frank, U. (2002). Multi-Perspective Enterprise Modeling (MEMO) - Conceptual Framework and Modeling Languages. In *Proceedings of the 35th Annual Hawaii International Conference on System Sciences* (pp. 1258--1267). IEEE.

Fritscher, B., & Pigneur, Y. (2011). Business IT Alignment from Business Model to Enterprise Architecture. In *Advanced Information Systems Engineering Workshops* (pp. 4--15). Springer.

Galster, M., Eberlein, A., & Moussavi, M. (2009). Comparing methodologies for the transition between software requirements and architectures. In *Proceedings of the International Conference on Systems, Man and Cybernetics* (pp. 2380--2385). IEEE.

Giaglis, G. M. (2001). A taxonomy of business process modeling and information systems modeling techniques. *International Journal of Flexible Manufacturing Systems*, *13*(2), 209--228.

Gigerenzer, G., & Gaissmaier, W. (2011). Heuristic decision making. *Annual Review of Psychology*, *62*, 451--482.

González, J. D. la V., & Díaz, J. (2007). Business process-driven requirements engineering: a goal-based approach. In *Proceedings of the 8th Workshop on Business Process Modeling, Development, and Support* (pp. 1--9). Tapir Academic Press.

Gröner, G., Asadi, M., Mohabbati, B., Gašević, D., Silva Parreiras, F., & Bošković, M. (2012). Validation of user intentions in process models. In *Advanced Information Systems Engineering* (pp. 366--381). Springer.

Grünbacher, P., Egyed, A., & Medvidovic, N. (2004). Reconciling software requirements and architectures with intermediate models. *Software & Systems Modeling*, *3*(3), 235--253.

Gyory, A., Brenner, W., & Uebernickel, F. (2012). Finding the Right Balanced Scorecard for Business-Driven IT Management A Literature Review. In *Proceedings of the 45th Hawaii International Conference on System Science* (pp. 5023--5032). IEEE.

Hansen, E. G., Grosse-Dunker, F., & Reichwald, R. (2009). Sustainability Innovation Cube — a Framework To Evaluate Sustainability-Oriented Innovations. *International Journal of Innovation Management*, *13*(4), 683--713.

Hansen, S., Berente, N., & Lyytinen, K. (2009). Requirements in the 21st Century: Current Practice and Emerging Trends. In *Design Requirements Engineering: A Ten-Year Perspective* (pp. 44--87). Springer.

Hasan, M. M., Loucopoulos, P., & Nikolaidou, M. (2014). Classification and Qualitative Analysis of Non-Functional Requirements Approaches. In *Enterprise, Business-Process and Information Systems Modeling* (pp. 348--362). Springer.

Henderson, J. C., & Venkatraman, N. (1993). Strategic alignment: Leveraging information technology for transforming organizations. *IBM SYSTEMS JOURNAL*, *32*(1), 4--16.

Henderson-Sellers, B., Zowghi, D., Klemola, T., & Parasuram, S. (2002). Sizing use cases: How to create a standard metrical approach. In *Object-Oriented Information Systems* (pp. 409--421). Springer.

Hevner, A., & Chatterjee, S. (2010). Design Science Research in Information Systems. In *Design Research in Information Systems* (pp. 9--22). Springer.

Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004). Design Science in Information Systems

Research. *MIS Quarterly*, *28*(1), 75--105.

Hickey, A., & Davis, A. (2004). A Unified Model of Requirements Elicitation. *Journal of Management Information Systems*, *20*(4), 65--84.

Horkoff, J., & Yu, E. (2011). Analyzing Goal Models – Different Approaches and How to Choose Among Them. In *Proceedings of the ACM Symposium on Applied Computing* (pp. 675--682). ACM.

Iacob, M., Mertens, L., Jonkers, H., Quartel, D. A. C., Nieuwenhuis, L. J. M., & Sinderen, M. J. van. (2012). From Enterprise Architecture to Business Models and back. *Software & Systems Modeling*, 1–25.

Iacob, M., Mertens, L., Jonkers, H., Quartel, D. A. C., Nieuwenhuis, L. J. M., & Sinderen, M. J. van. (2014). From Enterprise Architecture to Business Models and back. *Software and Systems Modeling*, *13*(3), 1059--1083.

Iivari, J., Hirschheim, R., & Klein, H. K. (2001). A dynamic framework for classifying information systems development methodologies and approaches. *Journal of Management Information Systems*, *17*(3), 179--218.

Jacobson, I., Booch, G., & Rumbaugh, J. (1999). *The unified software development process*. Addison-Wesley.

Jamshidi, P., Sharifi, M., & Mansour, S. (2008). To establish enterprise service model from enterprise business model. In *IEEE International Conference on Services Computing, SCC 2008* (Vol. 1, pp. 93–100).

Jarke, M., Loucopoulos, P., Lyytinen, K., Mylopoulos, J., & Robinson, W. (2011). The brave new world of design requirements. *Information Systems*, *36*(7), 992--1008.

Jensen, A. B. (2014). Do we need one business model definition? *Journal of Business Models*, *1*(1), 61--84.

Jonkers, H., Lankhorst, M., & Buuren, R. Van. (2004). Concepts for modeling enterprise architectures. *International Journal of Cooperative Information Systems*, *13*(3), 257--287.

Kaplan, R. S., & Norton, D. P. (1996). Linking the Balanced Scorecard to Strategy. *California Management Review*, *39*(1), 53--79.

Kapočius, K., & Butleris, R. (2006). Repository for business rules based IS requirements. *Informatica*, *17*(4), 503--518.

Kardasis, P., & Loucopoulos, P. (2005). A roadmap for the elicitation of business rules in information systems projects. *Business Process Management Journal*, *11*(4), 316--348.

Kazman, R., Klein, M., Barbacci, M., Longstaff, T., Lipson, H., & Carriere, J. (1998). The architecture tradeoff analysis method. In *Proceedings of the 4th International Conference on Engineering of Complex Computer Systems* (pp. 68--78). IEEE.

Kent, S. (2002). Model Driven Engineering. In *Integrated Formal Methods* (pp. 286--298). Springer.

Krogstie, J. (2008). Using EEML for Combined Goal and Process Oriented Modeling: A Case Study. In *Proceedings of the 13th International Workshop on Exploring Modeling Methods for Systems Analysis and Design* (pp. 112--129).

Kroll, P., & Kruchten, P. (2003). *The rational unified process made easy: a practitioner's guide to the RUP*. Addison-Wesley.

Lambert, S. (2006). A business model research schema. In *Proceedings of the 19th BLED eConference* (pp. 1--13). AIS.

Lamsweerde, A. Van. (2008). Requirements engineering: from craft to discipline. In *Proceedings of the 16th International Symposium on Foundations of Software Engineering* (pp. 238--249). ACM.

Lamsweerde, A. Van, & Willemet, L. (1998). Inferring declarative requirements specifications from operational scenarios. *IEEE Transactions on Software Engineering*, *24*(12), 1089--1114.

Lange, M., & Mendling, J. (2011). An experts' perspective on enterprise architecture goals, framework adoption and benefit assessmen. In *Proceedings of the 15th International Enterprise Distributed Object Computing Conference Workshops* (pp. 304--313). IEEE.

Lankhorst, M. (2004). Enterprise architecture modelling—the issue of integration. *Advanced Engineering Informatics*, *18*(4), 205--216.

Lapalme, J. (2012). Three schools of thought on enterprise architecture. *IT Professional*, *14*(6), 37--43.

Lee-Klenz, S., Sampaio, P., & Wood-Harper, T. (2010). A requirements elicitation framework and tool for sourcing business-IT aligned e-services. In *Proceedings of the ACM Symposium on Applied Computing* (pp. 111--117). ACM.

Loniewski, G., Insfran, E., & Abrahão, S. (2010). A Systematic Review of the Use of Requirements Engineering Techniques in Model-Driven Development. In *Model Driven Engineering Languages and Systems* (pp. 213--227). Springer.

López-Sanz, M., & Marcos, E. (2012). ArchiMeDeS: A model-driven framework for the specification of service-oriented architectures. *Information Systems*, *37*(3), 257--268.

Losavio, F., Chirinos, L., Matteo, A., Lévy, N., & Ramdane-Cherif, A. (2004). ISO quality standards for measuring architectures. *Journal of Systems and Software*, *72*(2), 209--223.

Losavio, F., Matteo, A., & Camejo, I. P. (2014). Unified Process for Domain Analysis integrating Quality , Aspects and Goals. *CLEI Electronic Journal*, *17*(2), 1--21.

Loucopoulos, P., & Garfield, J. (2009). The intertwining of enterprise strategy and requirements. In *Design Requirements Engineering: A Ten-Year Perspective* (pp. 352–373).

Lucena, M., Castro, J., Silva, C., Alencar, F., Santos, E., & Pimentel, J. (2009). A model transformation approach to derive architectural models from goal-oriented requirements models. In *On the Move to Meaningful Internet Systems: OTM 2009 Workshops* (pp. 370--380). Springer.

Luftman, J., Papp, R., & Brier, T. (1999). Enablers and inhibitors of business-IT alignment.

Communications of the AIS, *1*(March).

Machado, R. J., Fernandes, J., Monteiro, P., & Rodrigues, H. (2006). Refinement of Software Architectures by Recursive Model Transformations. In J. Münch & M. Vierimaa (Eds.), *Product-Focused Software Process Improvement* (pp. 422--428). Conference Paper, Springer Berlin / Heidelberg.

Maes, R., Rijsenbrij, D., Truijens, O., & Goedvolk, H. (2000). *Redefining business: IT alignment through a unified framework*. *Universiteit Van Amsterdam*.

Magjarevic, R. (2007). Home care technologies for ambient assisted living. In *Proceedings of the 11th Mediterranean Conference on Medical and Biomedical Engineering and Computing* (pp. 397--400). Springer.

Meertens, L. O., Iacob, M. E., Nieuwenhuis, L. J. M., Sinderen, M. J. Van, Jonkers, H., & Quartel, D. A. C. (2012). Mapping the Business Model Canvas to ArchiMate. In *Proceedings of the 27th Annual ACM Symposium on Applied Computing* (pp. 1694--1701). ACM.

Montero, F., & Navarro, E. (2009). ATRIUM: Software architecture driven by requirements. In *Proceedings of the 14th International Conference on Engineering of Complex Computer Systems* (pp. 230--239). IEEE.

Morris, M., Schindehutte, M., & Allen, J. (2005). The entrepreneur’s business model: toward a unified perspective. *Journal of Business Research*, *58*(6), 726--735.

Nuseibeh, B. (2001). Weaving together requirements and architectures. *Computer*, *34*(3), 115--119.

Nuseibeh, B., & Easterbrook, S. (2000). Requirements Engineering: A Roadmap. In *Proceedings of the Conference on The Future of Software Engineering* (pp. 35--46). ACM.

O’Brien, L., Merson, P., & Bass, L. (2007). Quality attributes for service-oriented architectures. In *Proceedings of the International Workshop on Systems Development in SOA Environments* (p. 3). IEEE.

Ostadzadeh, S. S. (2007). A Method for Consistent Modeling of Zachman Framework Cells. In *Advances and Innovations in Systems, Computing Sciences and Software Engineering* (pp. 375--380). Springer.

Osterwalder, A. (2004). *The business model ontology: A proposition in a design science approach*. University of Lausanne, Switzerland.

Osterwalder, A., & Pigneur, Y. (2010). *Business model generation: a handbook for visionaries, game changers, and challengers*. John Wiley and Sons.

Osterwalder, A., & Pigneur, Y. (2013). Designing Business Models and Similar Strategic Objects: The Contribution of IS. *Journal of the Association of Information Systems*, *14*(5), 237--244.

Osterwalder, A., Pigneur, Y., & Tucci, C. L. (2005). Clarifying business models: Origins, present, and future of the concept. *Communications of AIS*, *16*(1), 1--25.

Othman, R. (2008). Enhancing the effectiveness of the balanced scorecard with scenario planning. *International Journal of Productivity and Performance Management*, *57*(3), 259--

266.

Pateli, A. G., & Giaglis, G. M. (2004). A research framework for analysing eBusiness models. *European Journal of Information Systems*, *13*(4), 302--314.

Pépiot, G., Cheikhrouhou, N., Furbringer, J.-M., & Glardon, R. (2007). UECML: Unified Enterprise Competence Modelling Language. *Computers in Industry*, *58*(2), 130--142.

Persson, A., & Stirna, J. (2001). Why Enterprise modelling? An Explorative Study into Current Practice. In *Advanced Information Systems Engineering* (pp. 465--468). Springer.

Pieper, M., Antona, M., & Cortes, U. (2011). Ambient Assisted Living. *ERCIM News*, (87), 18--19.

Purao, S., Rossi, M., & Sein, M. K. (2010). On Integrating Action Research and Design Research. In *Design Research In Information Systems* (pp. 179--194). Springer.

Quartel, D., Engelsman, W., Jonkers, H., & Sinderen, M. Van. (2009). A Goal-Oriented Requirements Modelling Language for Enterprise Architecture. In *Proceedings of the 13th International Enterprise Distributed Object Computing Conference* (pp. 3--13). IEEE.

Rolland, C., Souveyet, C., & Achour, C. B. (1998). Guiding goal modeling using scenarios. *IEEE Transactions on Software Engineering*, *24*(12), 1055--1071.

Ruiz-Rube, I., Dodero, J. M., Palomo-Duarte, M., Ruiz, M., & Gawn, D. (2013). Uses and applications of Software & Systems Process Engineering Meta-Model process models. A systematic mapping study. *Journal of Software: Evolution and Process*, *25*(9), 999--1025.

Saat, J., Franke, U., Lagerström, R., & Ekstedt, M. (2010). Enterprise Architecture meta models for IT/business alignment situations. In *Proceedings of the 14th IEEE International Enterprise Distributed Object Computing Workshop* (pp. 14--23). IEEE.

Salgado, C. E., Machado, R. J., & Maciel, R. S. P. (2014a). An OMG-based meta-framework for alignment of IS/IT Architecture with Business Models. In *Proceedings of the 9th International Conference on the Quality of Information and Communications Technology* (pp. 288--293). IEEE.

Salgado, C. E., Machado, R. J., & Maciel, R. S. P. (2014b). Using Process-level Use Case Diagrams to Infer the Business Motivation Model with a RUP-based Approach. In *Information System Development* (pp. 123--134). Springer.

Salgado, C. E., Machado, R. J., & Maciel, R. S. P. (2015a). A Three-dimensional, Requirements-based, Balanced Scorecard Business Model. In *Proceedings of the 6th International Conference on Information and Communication Systems* (pp. 268--273). IEEE.

Salgado, C. E., Machado, R. J., & Maciel, R. S. P. (2015b). Aligning Business Requirements with Services Quality Characteristics by Using Logical Architectures. In *New Contributions in Information Systems and Technologies* (pp. 593--602). Springer.

Salgado, C. E., Machado, R. J., & Maciel, R. S. P. (2015c). Exploring a Three-Dimensional, Requirements-Based, Balanced Scorecard Business Model: On the Elicitation and Generation of a Business Model Canvas. In *Proceedings of the 17th Conference on Business Informatics* (pp. 88--95). IEEE.

Salgado, C. E., Machado, R. J., & Maciel, R. S. P. (2016a). A Three-dimensional Approach for a Quality-based Alignment between Requirements and Architecture. In *Exploring Services Science* (pp. 1--14). Springer.

Salgado, C. E., Machado, R. J., & Maciel, R. S. P. (2016b). MATRA : A Framework for Assessing Model-based Approaches on the Transformation between Requirements and Architecture. In *Enterprise, Business-Process and Information Systems Modeling* (pp. 1--12). Springer.

Salgado, C. E., Teixeira, J., Machado, R. J., & Maciel, R. S. P. (2014a). Generating a Business Model Canvas through Elicitation of Business Goals and Rules from Process-level Use Cases. In *Perspectives in Business Informatics Research* (pp. 276--289). Springer.

Salgado, C. E., Teixeira, J., Machado, R. J., & Maciel, R. S. P. (2014b). Generating a Business Model through the Elicitation of Business Goals and Rules within a SPEM approach. In *Information and Software Technologies* (pp. 47--58). Springer.

Salgado, C. E., Teixeira, J., Santos, N., Machado, R. J., & Maciel, R. S. P. (2015). A SoaML Approach for Derivation of a Process-Oriented Logical Architecture from Use Cases. In *Exploring Services Science* (pp. 80--94). Springer.

Salgado, C., Machado, R. J., & Suzana, R. (2013). Modeling the alignment between business and IS/IT: a requirements engineering perspective. In *Proceedings of the 28th Annual ACM Symposium on Applied Computing* (pp. 1309--1311). ACM.

Santander, V., & Castro, J. (2002). Deriving use cases from organizational modeling. In *Proceedings of the Joint International Conference on Requirements Engineering* (pp. 32--39). IEEE.

Santos, N., Teixeira, J., Pereira, A., Ferreira, N., Lima, A., Simões, R., & Machado, R. J. (2012). A Demonstration Case on the Derivation of Process-Level Logical Architectures for Ambient Assisted Living Ecosystems. In *Ambient Assisted Living Book*. CRC.

Self, J. (2004). Metrics and management: applying the results of the balanced scorecard. *Performance Measurement and Metrics*, *5*(3), 101--105.

Selic, B. (2012). What will it take? A view on adoption of model-based methods in practice. *Software & Systems Modeling*, *11*(4), 513--526.

Shanmugapriya, P., & M. Suresh, R. (2012). Software Architecture Evaluation Methods - A Survey. *International Journal of Computer Applications*, *49*(16), 19--26.

Shao, J., & Pound, C. (1999). Extracting business rules from information systems. *BT Technology Journal*, *17*(4), 179--186.

Simon, D., Fischbach, K., & Schoder, D. (2013). An exploration of enterprise architecture research. *Communications of the Association for Information Systems*, *32*(1), 1--72.

Slaughter, S. A., Levine, L., Ramesh, B., Pries-Heje, J., & Baskerville, R. (2006). Aligning software processes with strategy. *MIS Quarterly*, *30*(4), 891--918.

Solingen, R. van, Basili, V., Caldiera, G., & Rombach, H. D. (2002). Goal Question Metric (GQM) Approach. In *Encyclopedia of Software Engineering*. Wiley.

Sowa, J. F., & Zachman, J. A. (1992). Extending and formalizing the framework for information systems architecture. *IBM Systems Journal*, *31*(3), 590–-616.

Stirna, J., Grabis, J., Henkel, M., & Zdravkovic, J. (2012). Capability Driven Development – An Approach to Support Evolving Organizations. In *The Practice of Enterprise Modeling* (pp. 117--131). Springer.

Stroian, V., Bourque, P., & Abran, A. (2006). Visualization: A Key Concept for Multidimensional Performance Modeling in Software Engineering Management. In *Proceedings of the International Conference on Automation, Quality and Testing, Robotics* (pp. 334--339). IEEE.

Supakkul, S., & Chung, L. (2005). A UML profile for goal-oriented and use case-driven representation of NFRs and FRs. In *Proceedings of the 3rd ACIS International Conference on Software Engineering Research, Management and Applications* (pp. 112--119). IEEE.

Sutcliffe, A., & Sawyer, P. (2013). Requirements elicitation: Towards the unknown unknowns. In *Proceedings of the 21st International Requirements Engineering Conference* (pp. 92--104). IEEE.

Tamm, T., Seddon, P., Shanks, G., & Reynolds, P. (2011). How Does Enterprise Architecture Add Value to Organisations? *Communications of the Association for Information Systems*, *28*(1), 141--168.

Terävä, H. (2007). *Software Process Modeling with Eclipse Process Framework and SPEM 2.0*. University of Turku.

Touzi, J., Benaben, F., Pingaud, H., & Lorré, J. P. (2009). A model-driven approach for collaborative service-oriented architecture design. *International Journal of Production Economics*, *121*(1), 5--20.

Ullah, A., & Lai, R. (2011). Modeling business goal for business/it alignment using requirements engineering. *Journal of Computer Information Systems*, *51*(3), 21--28.

Ullah, A., & Lai, R. (2013). A Systematic Review of Business and Information Technology Alignment. *ACM Transactions on Management Information Systems*, *4*(1), 1--30.

Vernadat, F. (2002). UEML: towards a unified enterprise modelling language. *International Journal of Production Research*, *40*(17), 4309--4321.

Wan-Kadir, W. M., & Loucopoulos, P. (2004). Relating evolving business rules to software design. *Journal of Systems Architecture*, *50*(7), 367--382.

Whittle, J., Hutchinson, J., & Rouncefield, M. (2014). The State of Practice in Model-Driven Engineering. *IEEE Software*, *31*(3), 79--85.

Wieringa, R. (2004). Requirements engineering: Problem analysis and solution specification. In *Web Engineering* (pp. 13--16). Springer.

Winkler, S., & von Pilgrim, J. (2010). A survey of traceability in requirements engineering and model-driven development. *Software & Systems Modeling*, *9*(4), 529--565.

Yamamoto, S., Kaiya, H., Karl, C., & Bleistein, S. (2006). Goal oriented requirements engineering: trends and issues. *IEICE TRANSACTIONS on Information and Systems*, *89*(11), 2701--2711.

Yu, E., & Mylopoulos, J. (1994). Using Goals, Rules, and Methods to Support Reasoning in Business Process Reengineering. In *Proceedings of the 27th Hawaii International Conference on System Sciences* (pp. 234--243). IEEE.

Yu, E. S. K. (1997). Towards modelling and reasoning support for early-phase requirements engineering. *Proceedings of the 3rd International Symposium on Requirements Engineering*, 226--235.

Yu, E. S. K., & Mylopoulos, J. (1994). Using Goals, Rules, and Methods to Support Reasoning in Business Process Reengineering. In *Proceedings of the Twenty-Seventh Hawaii International Conference on System Sciences* (pp. 234–243).

Yu, E., Strohmaier, M., & Deng, X. (2006). Exploring Intentional Modeling and Analysis for Enterprise Architecture. In *Procedings of the 10th IEEE International Enterprise Distributed Object Computing Conference Workshops* (pp. 32--39). IEEE.

Yue, T., Briand, L. C., & Labiche, Y. (2011). A systematic review of transformation approaches between user requirements and analysis models. *Requirements Engineering*, *16*(2), 75--99.

Zachman, J. a. (1987). A framework for information systems architecture. *IBM Systems Journal*, *26*(3), 276--292.

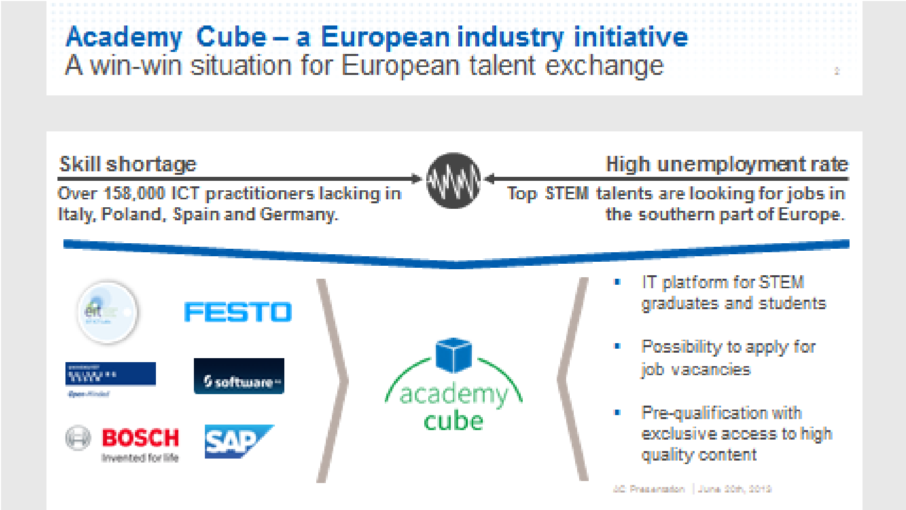
Zimmermann, A., Pretz, M., Zimmermann, G., Firesmith, D. G., Petrov, I., & El-Sheikh, E. (2013). Towards service-oriented enterprise architectures for big data applications in the cloud. In *Proceedings of the 17th International Enterprise Distributed Object Computing Conference Workshops* (pp. 130--135). IEEE.

Zott, C., Amit, R., & Massa, L. (2011). The Business Model: Recent Developments and Future Research. *Journal of Management*, *37*(4), 1019–1042.

Zowghi, D., & Coulin, C. (2005). Requirements Elicitation: A Survey of Techniques, Approaches and Tools. In *Engineering and managing software requirements* (pp. 19--46). Springer.

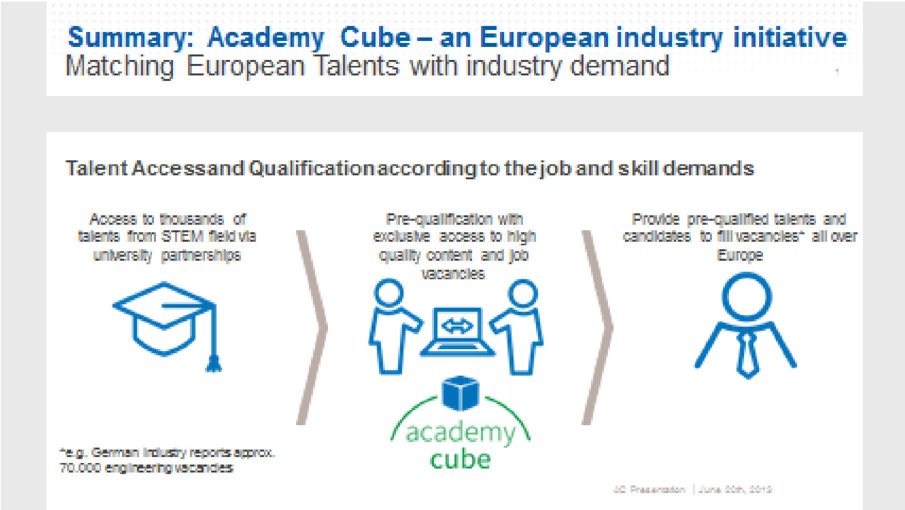
Appendix A

In this appendix, we present the project 'Academic Cube' (AC.pt), along with the use of the PGR and PGR-BSC Modeler tools:

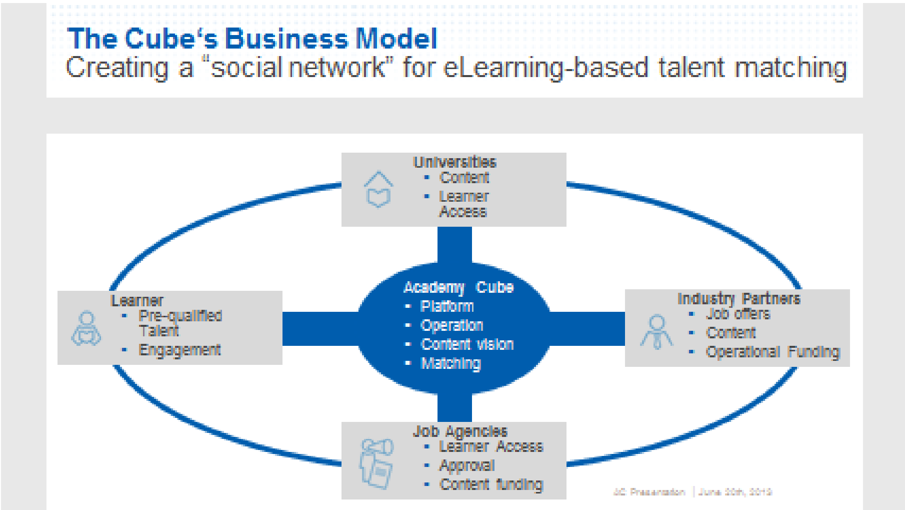


This project counted with numerous relevant partners and supporters, with special relevance to SAP and BOSCH:

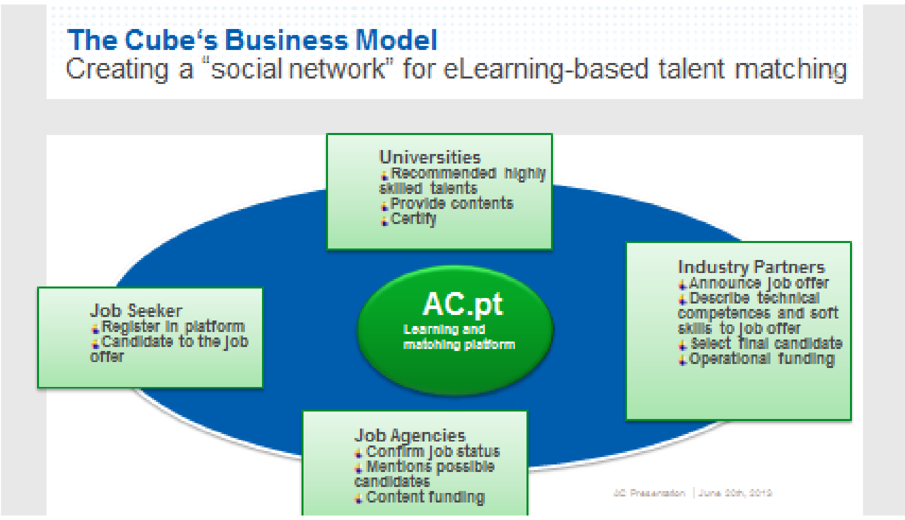
It is a project framed within a European industry initiative:



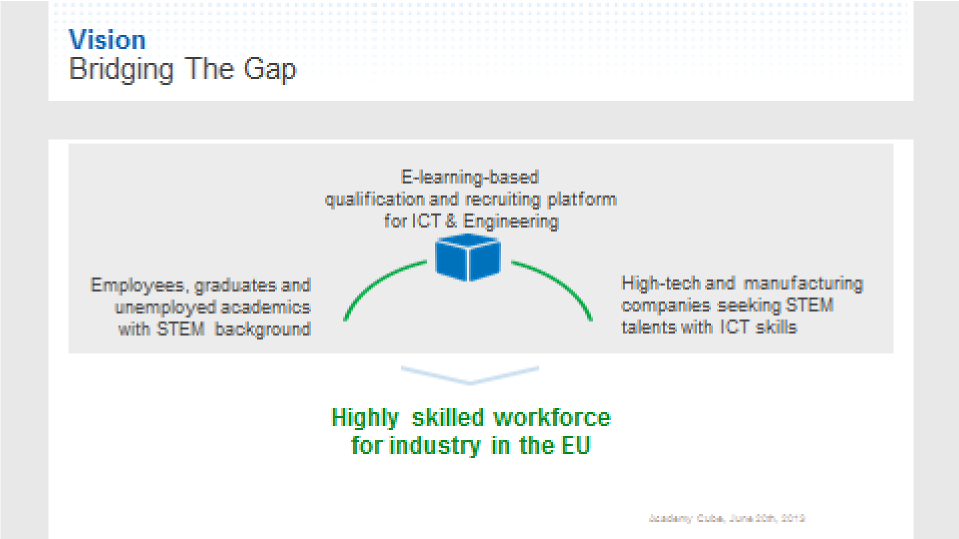
The initial business model pointed to four vectors, namely Universities, Learners, Job Agencies and Industry Partners:

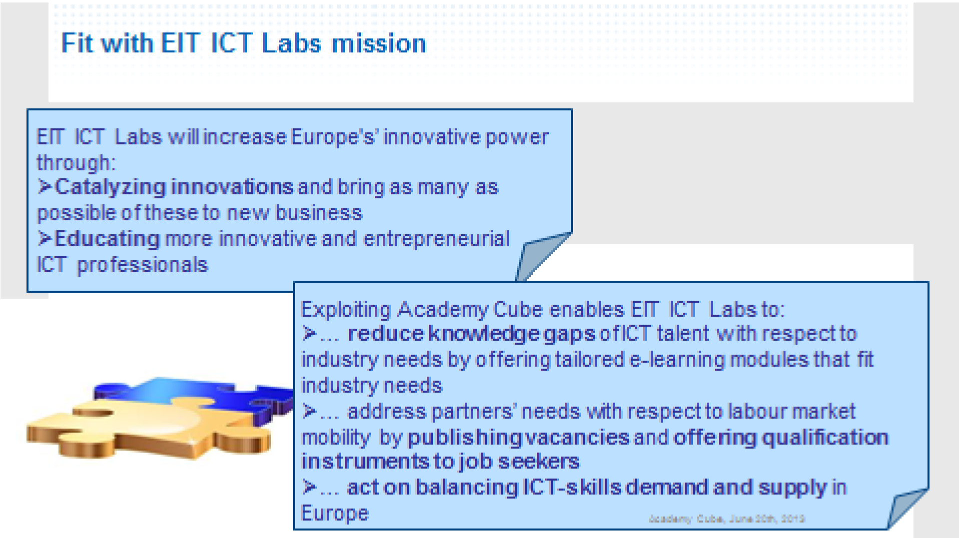


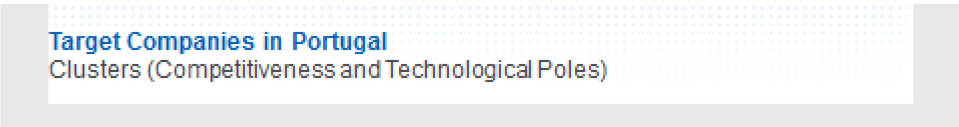
Initially, there was no specific method for achieving a consensual business model:



Nevertheless, there was an established vision, and sparse clustered objectives and strategies:







Poles	Objectives/Strategy
Engineering and Tooling	✓ To place the Engineering & Tooling sector in the world's top 5 sectors ✓ To have new technology-based enterprises operating in the sector ✓ To achieve high levels of recognition for the Engineering & Tooling Portugal brand, at national and international ✓ To ensure increasing annual investment in the areas of research, development and innovation ✓ To **attract qualified personnel** to the Enterprises
Mobility Industries	✓ The strategy focuses on the confluence of the three base industries - automotive, aeronautics and railways - and seeks new industrial paradigms in cooperation with the energy sector and the ECIT (Electronics, Communication and Information Technologies). ✓ To develop a suppliers industry that is strong and **skilful** and that demonstrates **excellence in performance** ✓ To increase the level of **highly-qualified employment** in the enterprises of the mobility industries
Refining, Petrochemical and Industrial Chemical	✓ Attract leading international investment companies, to manufacture those products in which Portugal is not yet self-sufficient and to create innovation, Technology and Development Centres operating as a network ✓ To ensure that the Centre's personnel develop the **required skills**

Target Companies in Portugal

Clusters (Competitiveness and Technological Poles)

Poles	Objectives/Strategy
PRODUTECH - Production Technologies	✓ Enhance national competencies and skills; ✓ Project a unified image; ✓ Foster research by establishing privileged contacts with higher education institutions, research units and technology centres. ✓ To increase private investment in R&TD (Research and Technology Development) and innovation as well as its efficiency and effectiveness, through the creation or reinforcement of R&TD management skills within the enterprises involved and the adoption of good practices in the conduct of relationships with the SCTN (National Science and Technology System) entities
Energy	✓ To support R&TD activities in Portugal, particularly as concerns renewable energy and energy-efficient technologies and systems associated with international projects ✓ To **create skilled employment**, namely in the areas of research, technical support and promotional activities ✓ To improve the skills of the personnel of enterprises in the sector

Academy Cube, June 20th, 2013

Objectives in Portugal 2013 -2016

An expanding job market for highly skilled ICT labours for Portuguese industrial partners

Regular replenishment of industrial partners: ~5000 learners per year (35 partners)

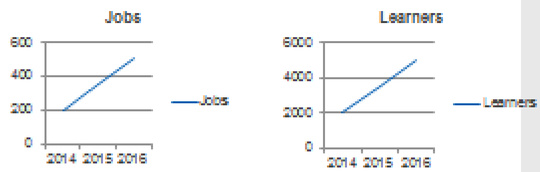
Extra: Need for new jobs and skills due to industrial and ICT developments (Industry 4.0)

Extra: use of Academy Cube platform for other purposes as a repository for professional eLearning education

Target Academy Cube to Portugal (2013-2016):

National Job placements – 1000

Learners - 10000

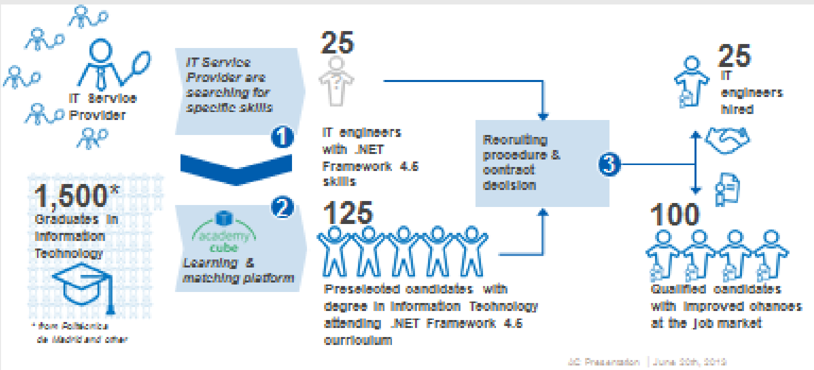


Academy Cube, June 20th, 2013

Its associated questions, data and expectations were high:

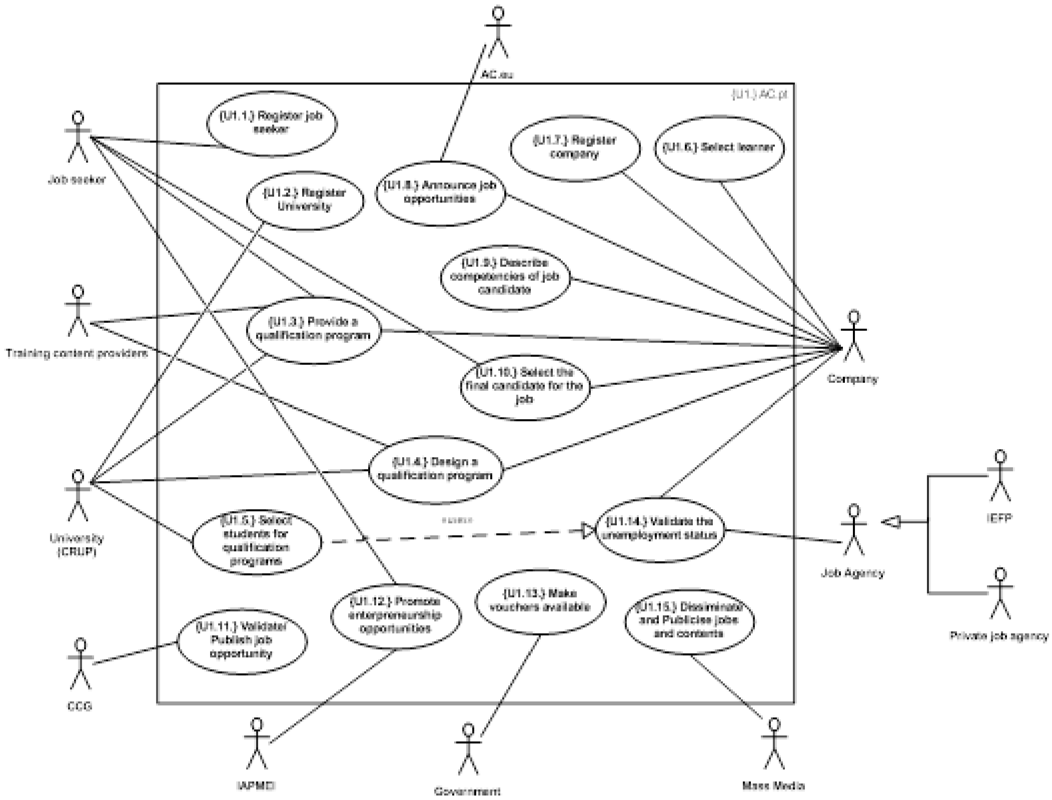
Match with highly skilled talents

How can you accelerate your recruiting outputs?

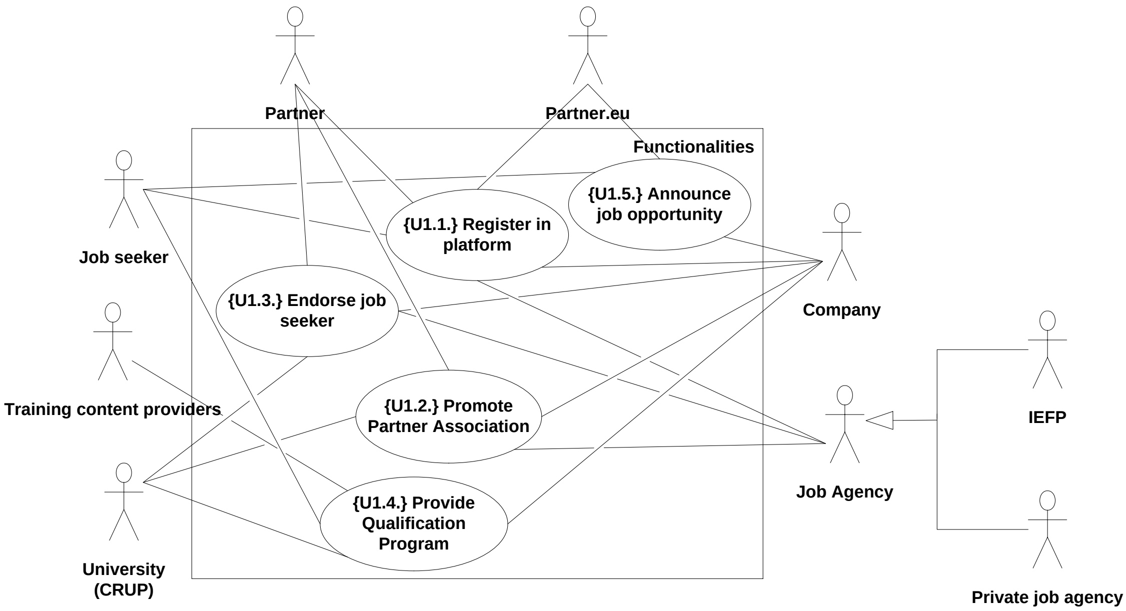


AC Presentation | June 20th, 2013

The initial solution design involved ten actors interacting with fifteen Use-cases:

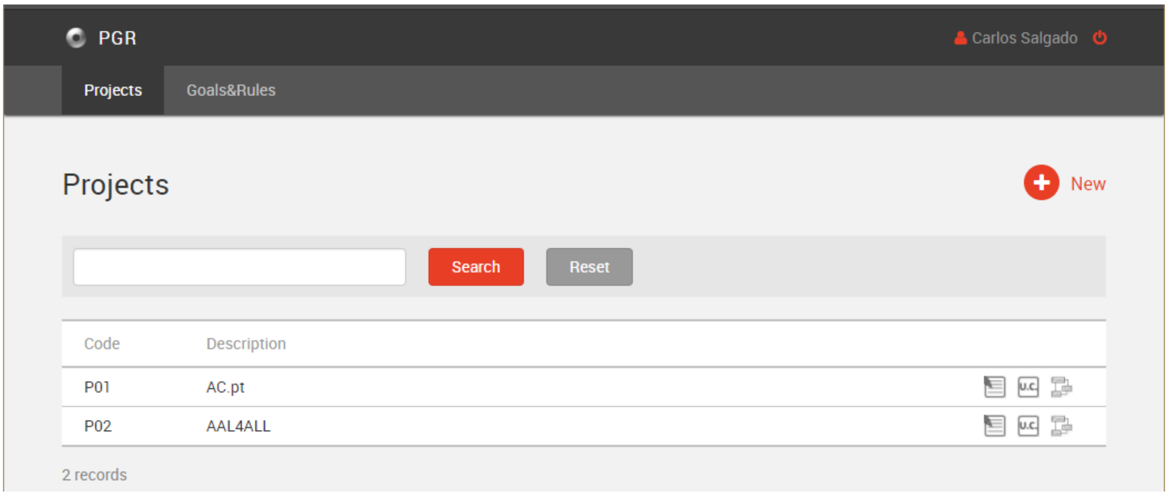


A condensed scenario was selected to endorse and develop the project:

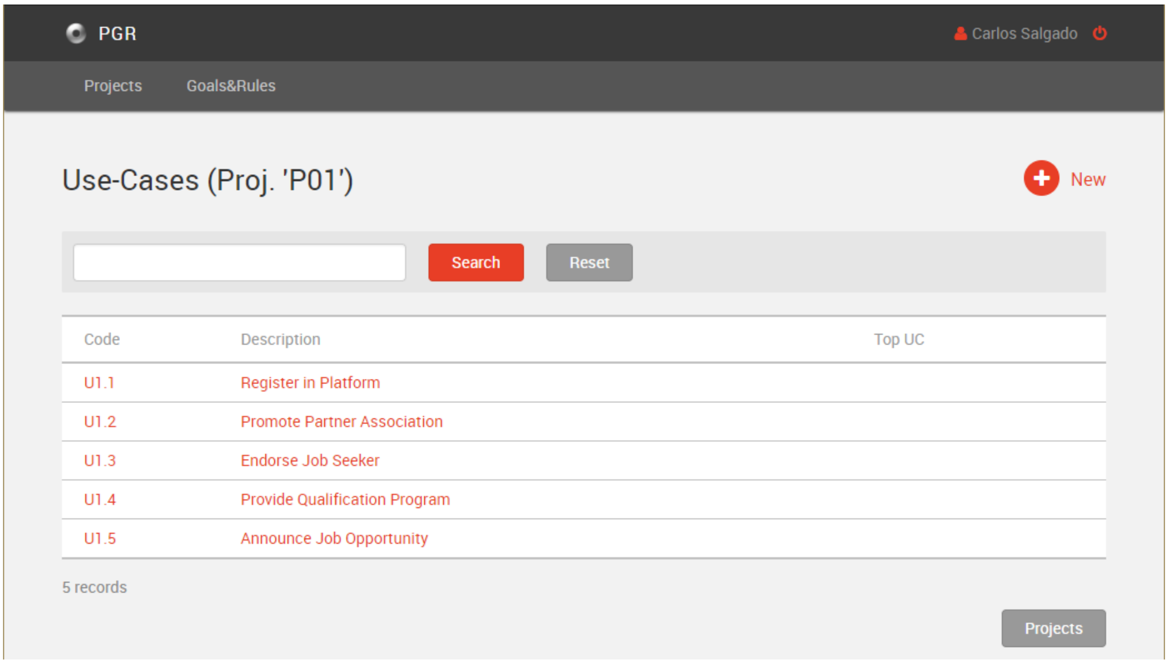


The project data, according to the methods followed (PGR and PGR-BSC), was used to test and demonstrate the associated prototype tools.

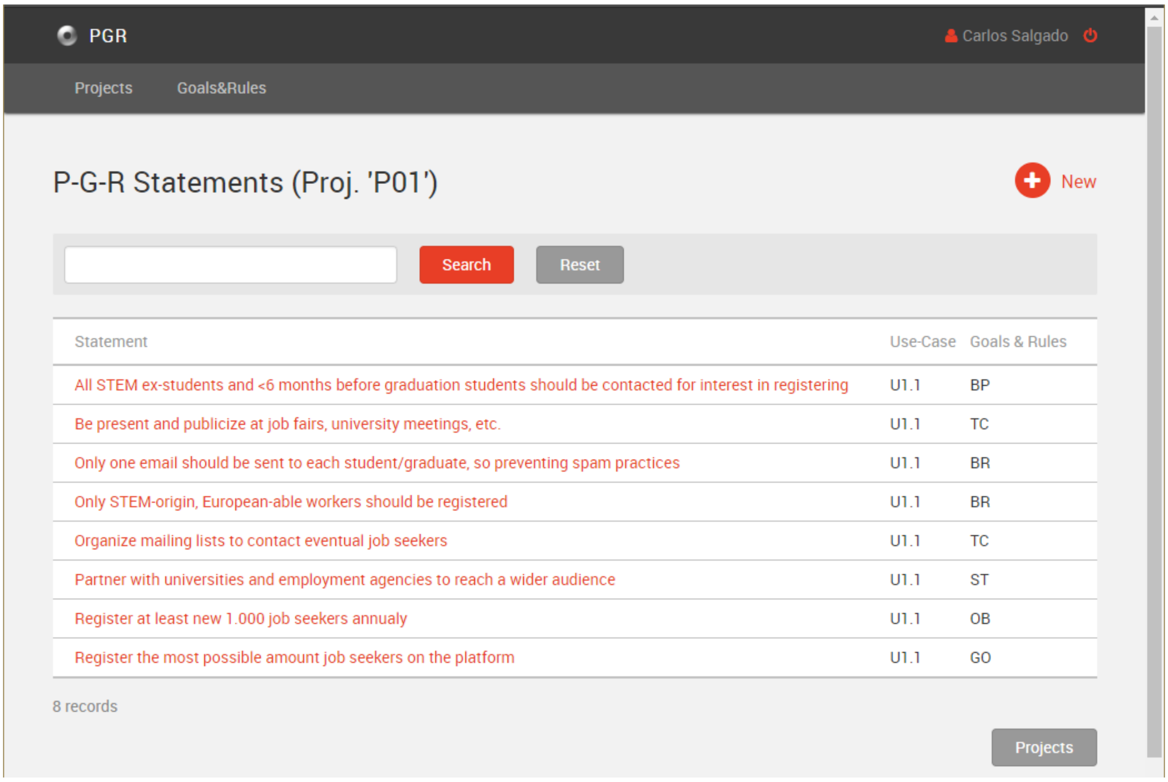
[PGR] Projects listing:



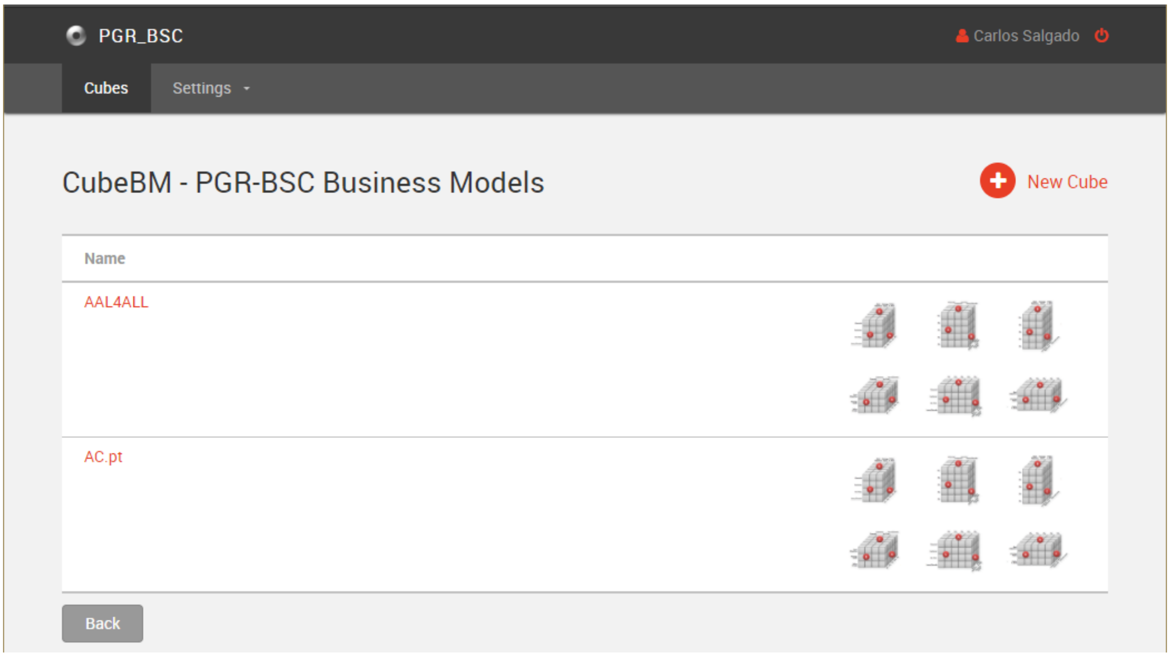
[PGR] List of Use-cases associated to a Project:



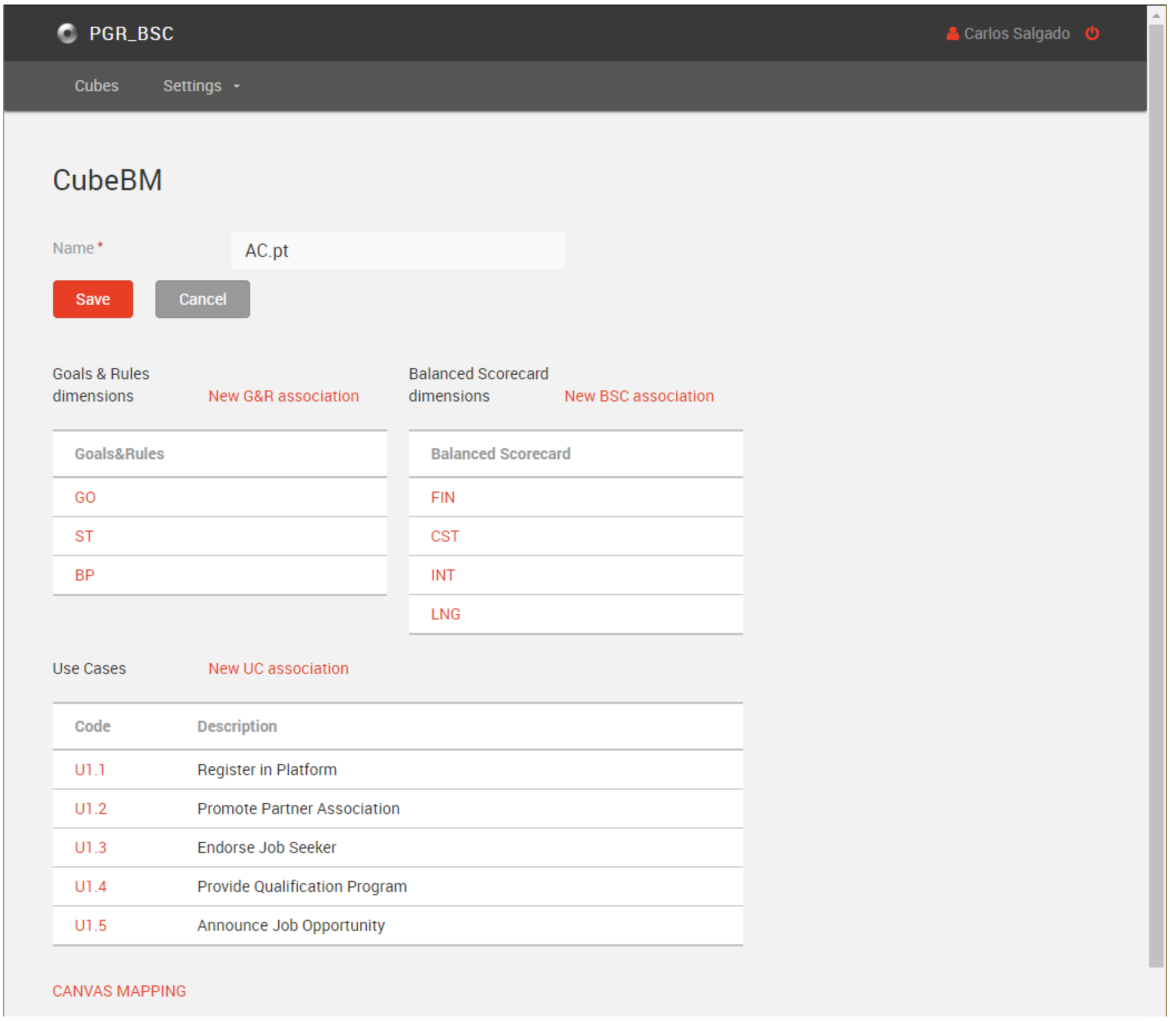
[PGR] List of PGR Statements associated to a Project:



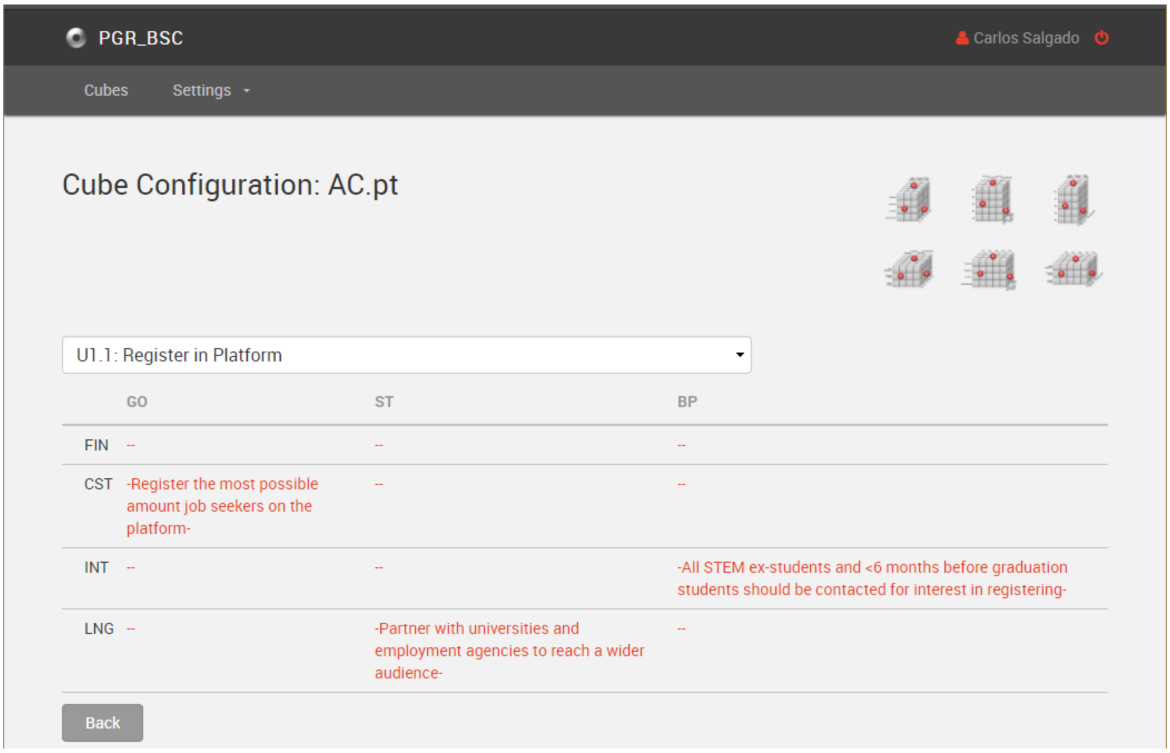
[PGR-BSC] List of PGR-BSC Cubes:



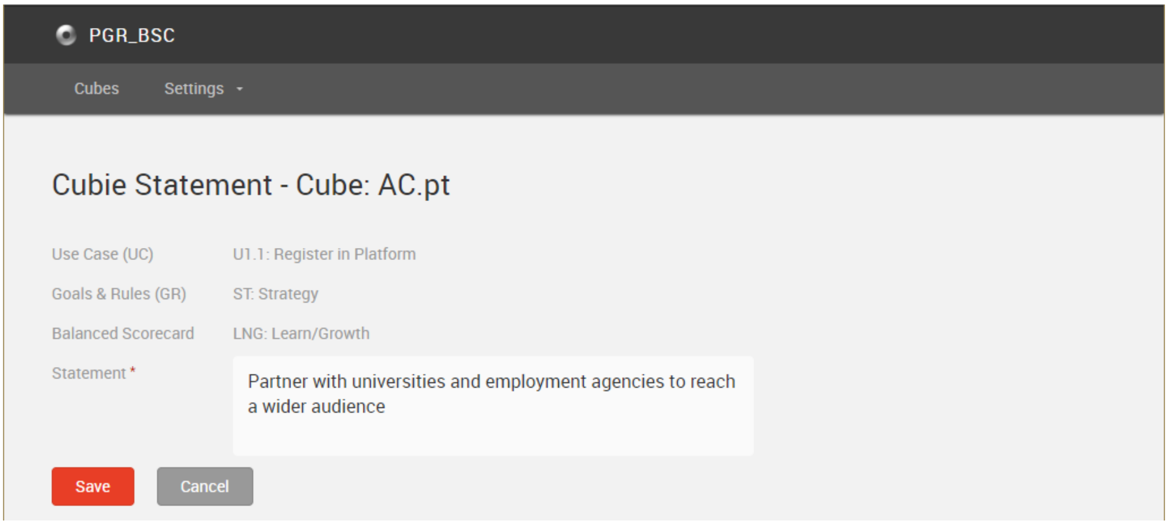
[PGR-BSC] Detail of a PGR-BSC Cube Business Model with respective associations (G&R, BSC, UC):



[PGR-BSC] Use-case Perspective of a PGR-BSC Cube:



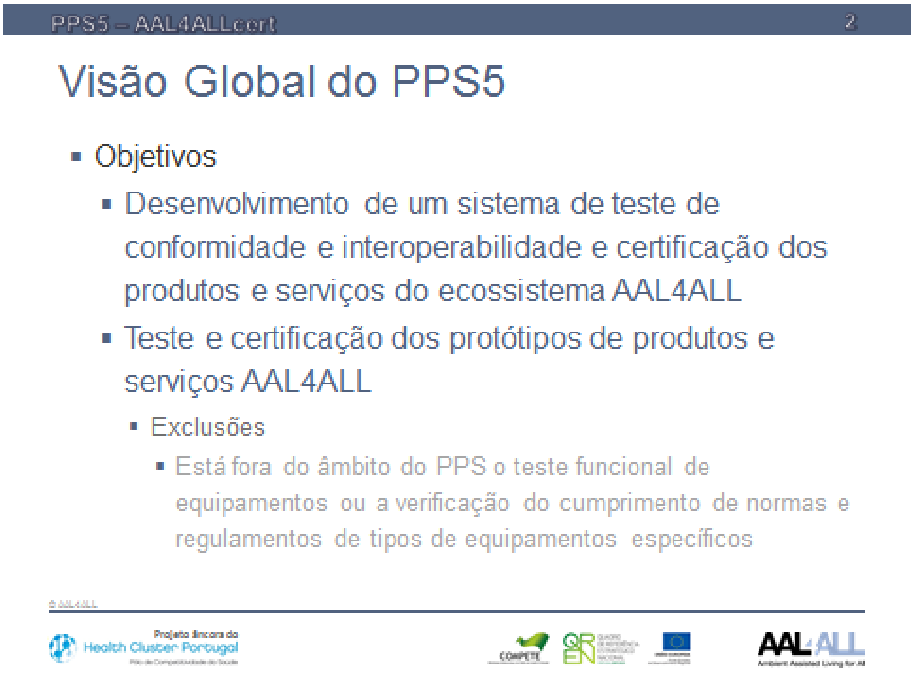
[PGR-BSC] Detail of a PGR-BSC Cubie Statement:



Appendix B

In this appendix, we present the project ‘AAL4ALL’, along with the corresponding use of the PGR-BSC and PGR-SOA Modeler tools.

The work related to this project stands within the scope of PPS5 of the AAL4ALL project, which aims to develop a system of compliance and interoperability testing and certification of products and services of the AAL4ALL ecosystem, framed in the task T5.1.1. "Definition of the business model and management of AAL4ALL testing and certification activity". Its main objectives are the definition of business model of the Certification Body and the current management of this entity, supporting the activities of standardization, testing and certification.

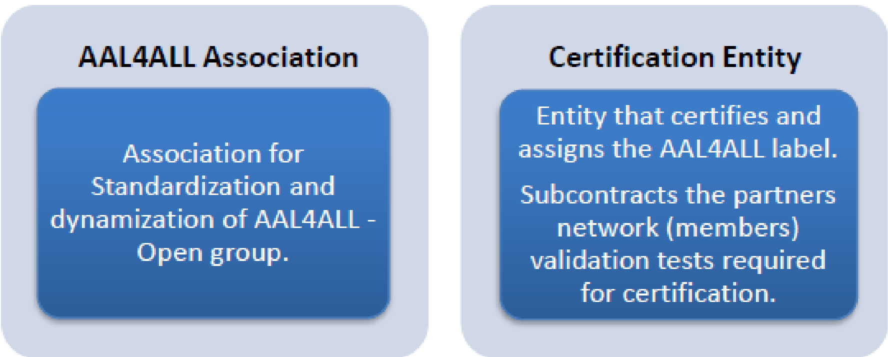


This project counted with numerous relevant partners and supporters, with special relevance to CITEVE and Microsoft:

One of the deliverables is related to its ecosystem validation and certification system, with the definition of a business model, and of test and certification management activities:

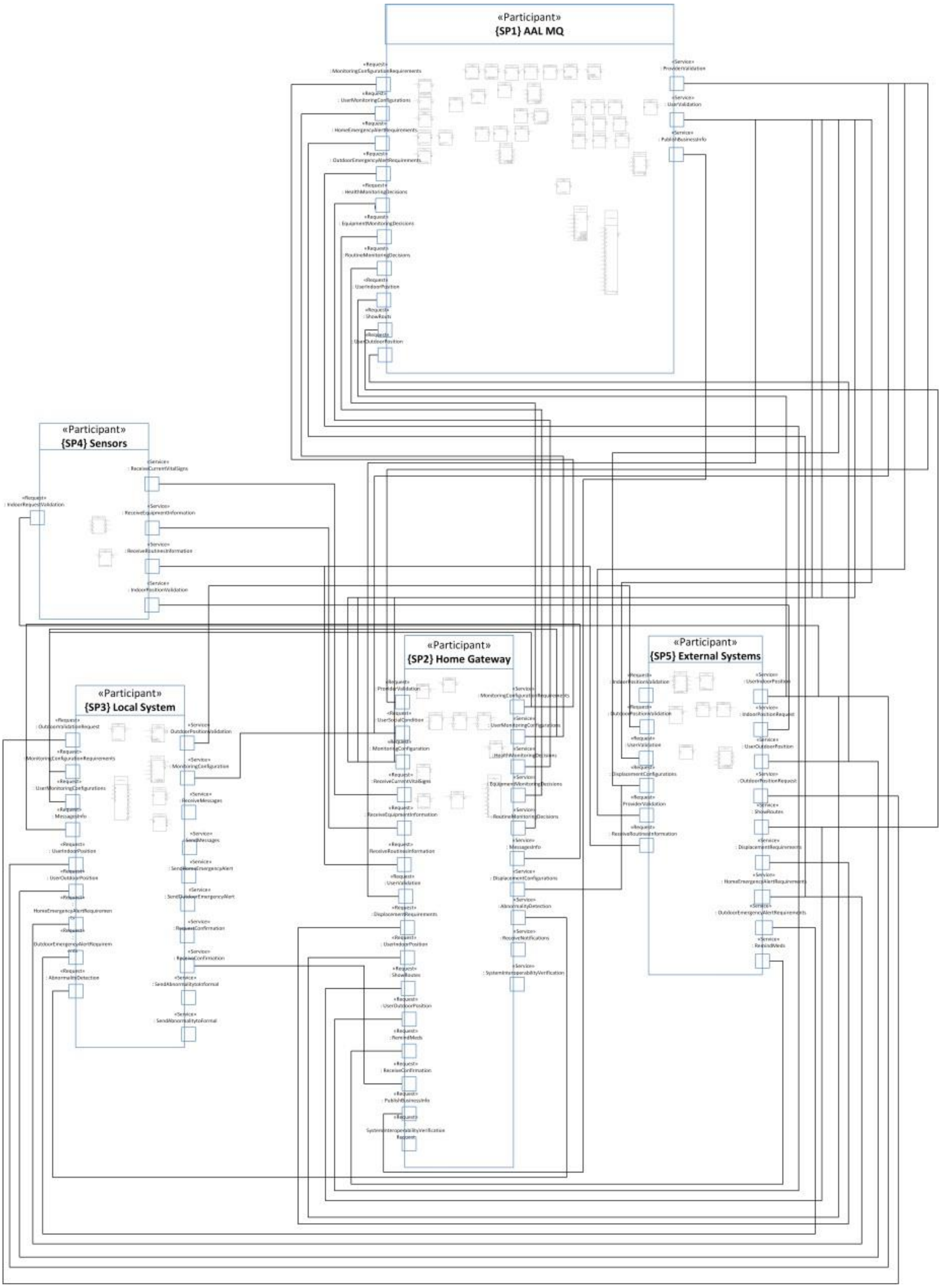
Key Partners	Key Activities	Value Proposition	Customer relationship	Customer Segments
Key organizations representing the demand (National Level and European Level) Universities (National Level and European Level) RTD entities (National Level and European Level) AAL4ALL Association	1. Test and certification activities (issuing, renewal, upgrade, ...) 2. Development of know-how 3. Standard Implementation 4. Validation mechanisms - Auditing 5. Marketing activities 6. Certification services	A Label for accelerating the Commercialization of the AAL Solutions Assuring the ability of a standard for development	Direct Proximity Partnership	Companies with AAL4ALL Solutions Members of AAL4ALL Association
	Key Resources		**Channels**	
	Management structure Support structure Marketing Structure		Direct Chanel (Web service) International Fairs	
Cost Structure			**Revenue Streams**	
Certification Costs Development of know-how and Standard Implementation Marketing Costs			Certification costs Training (technicians)	

These deliverables are aimed at a vision of an organizational model within a double orthogonal reality, the AAL4ALL Association and its respective Certification Entity:

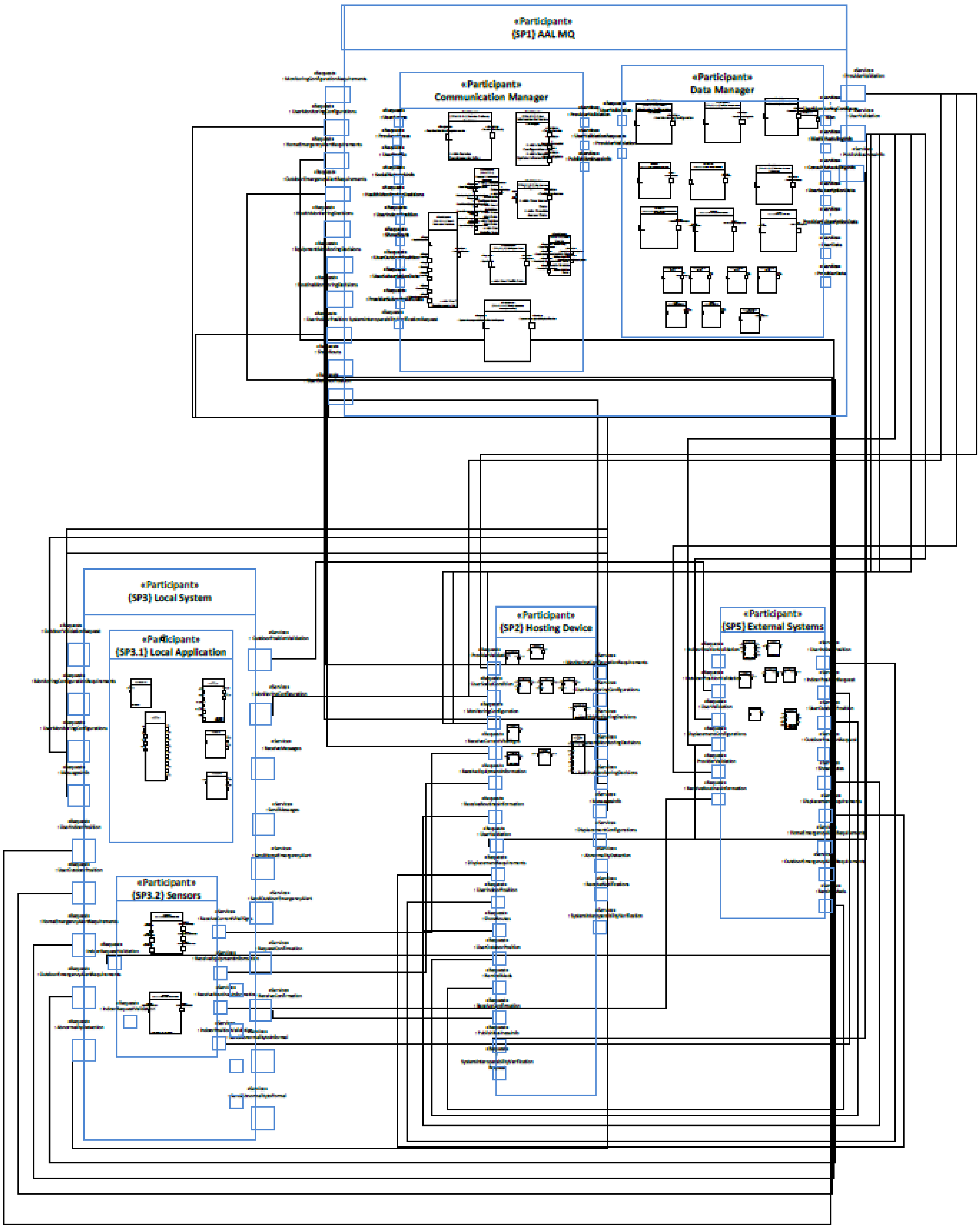


Both the AAL4ALL Association and the AAL4ALL Certification Entity had its proper organizational and functional directives, according to each Entity responsabilities:

Interoperability needs within Services and Products of the AAL4ALL ecosystem led to an evolution of the traditional logical architecture from the 4SRS, taking into account diverse business issues (nonfunctional), this time with the 4SRS-SoaML:

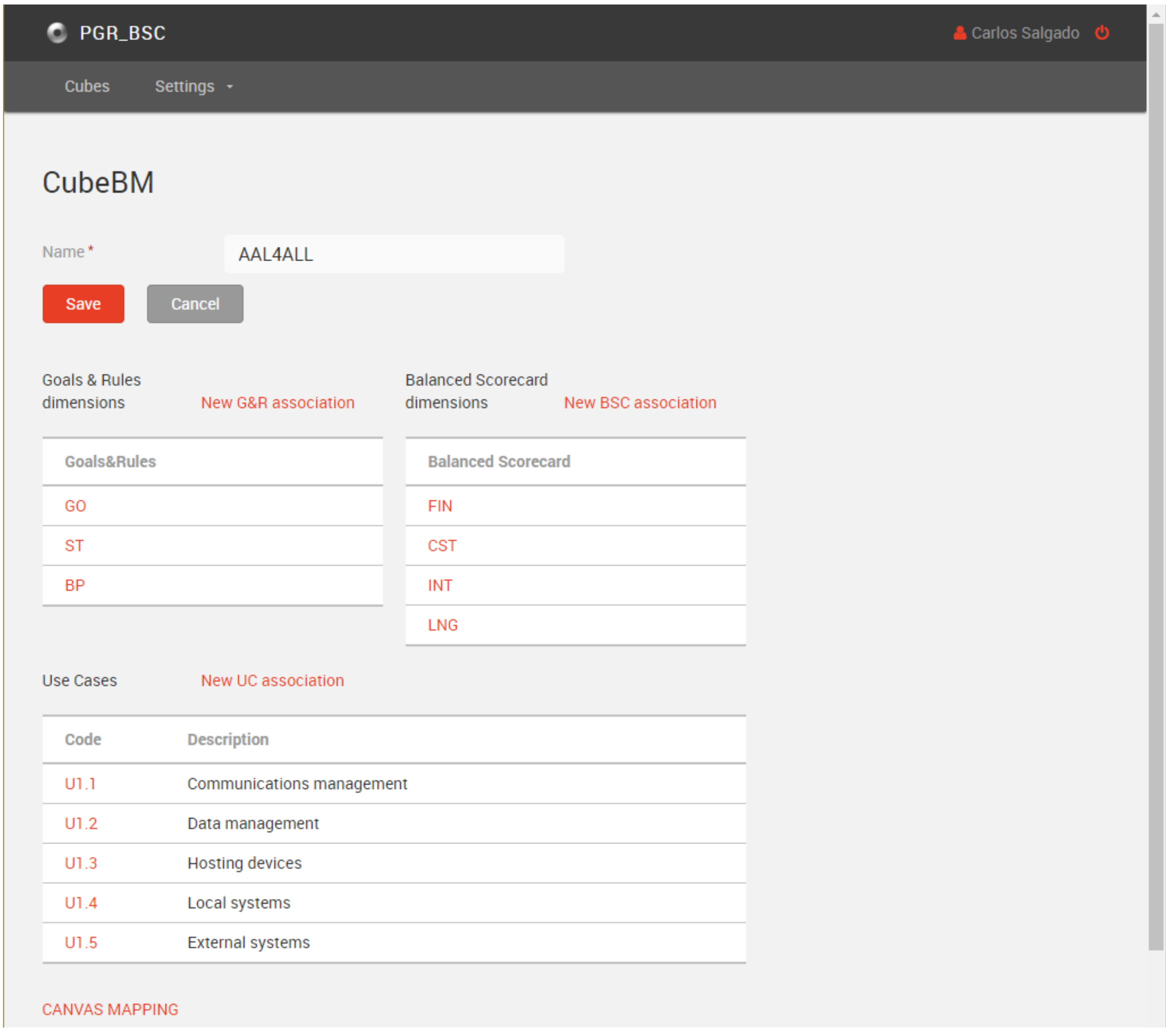


Following, in a second iteration of this method, a new version of the architecture was achieved in order to support the aggregated communication channels of each of the individual architectural elements of the ecosystem:

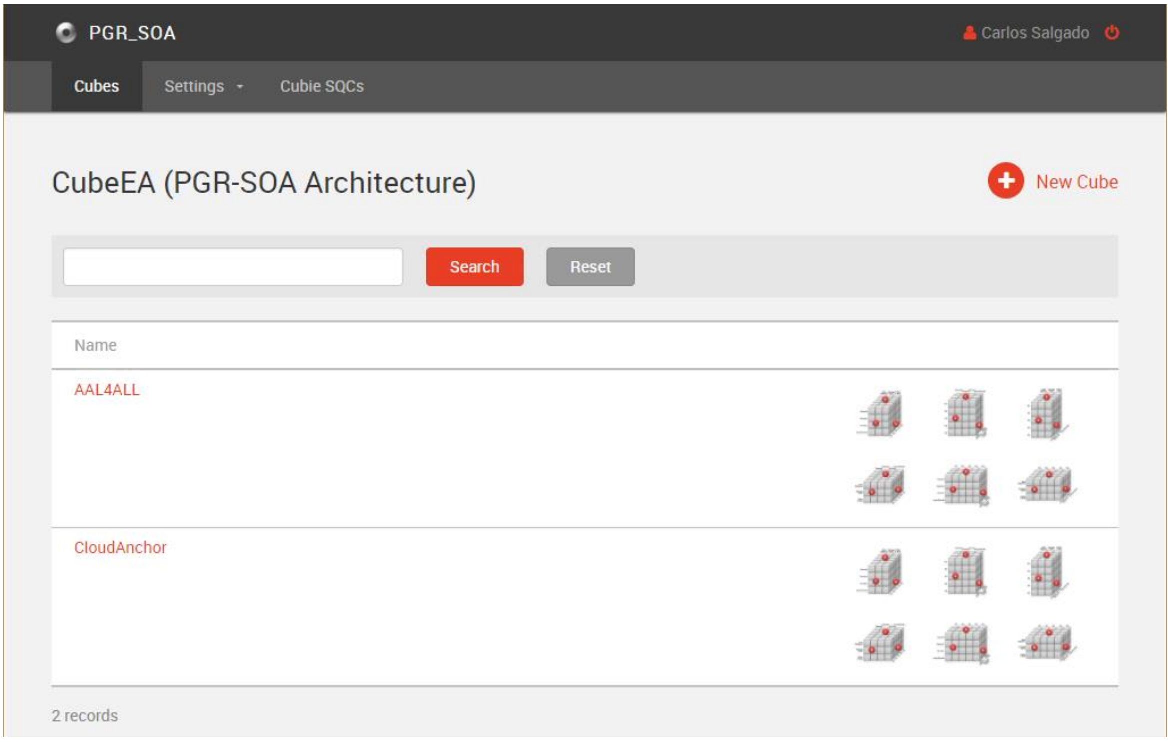


The project data, according to the methods followed (PGR-BSC and PGR-SOA), was used to test and demonstrate the associated prototype tools.

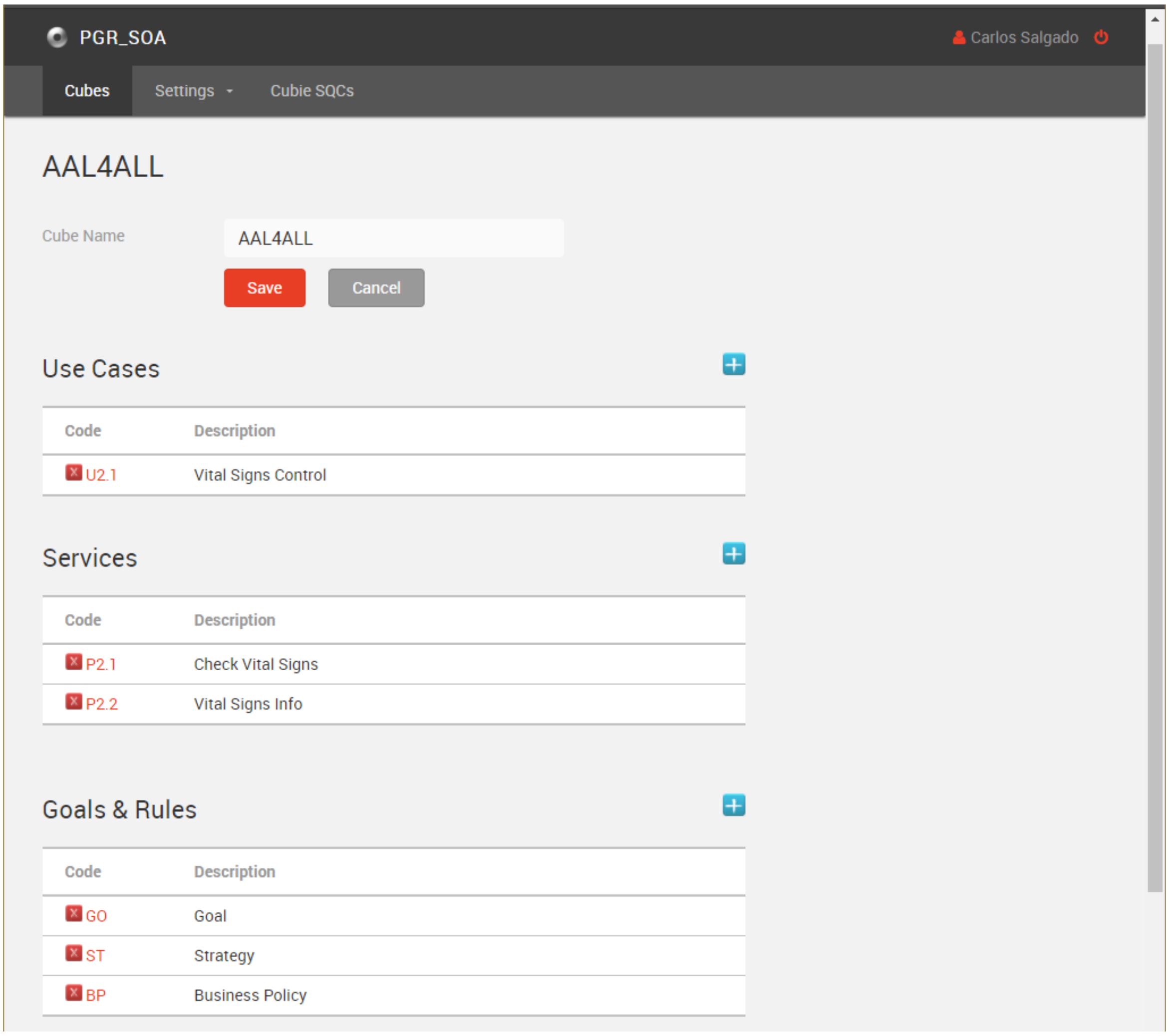
[PGR-BSC] Detail of a PGR-BSC Cube Business Model with respective associations (G&R, BSC, UC):



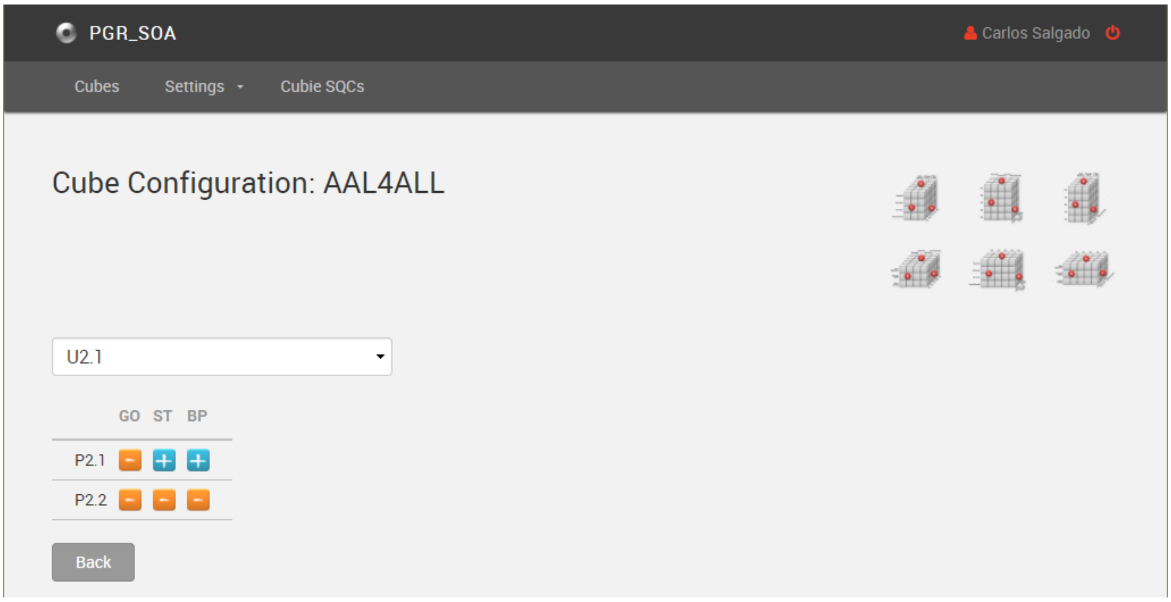
[PGR-SOA] List of PGR-SOA Cubes:



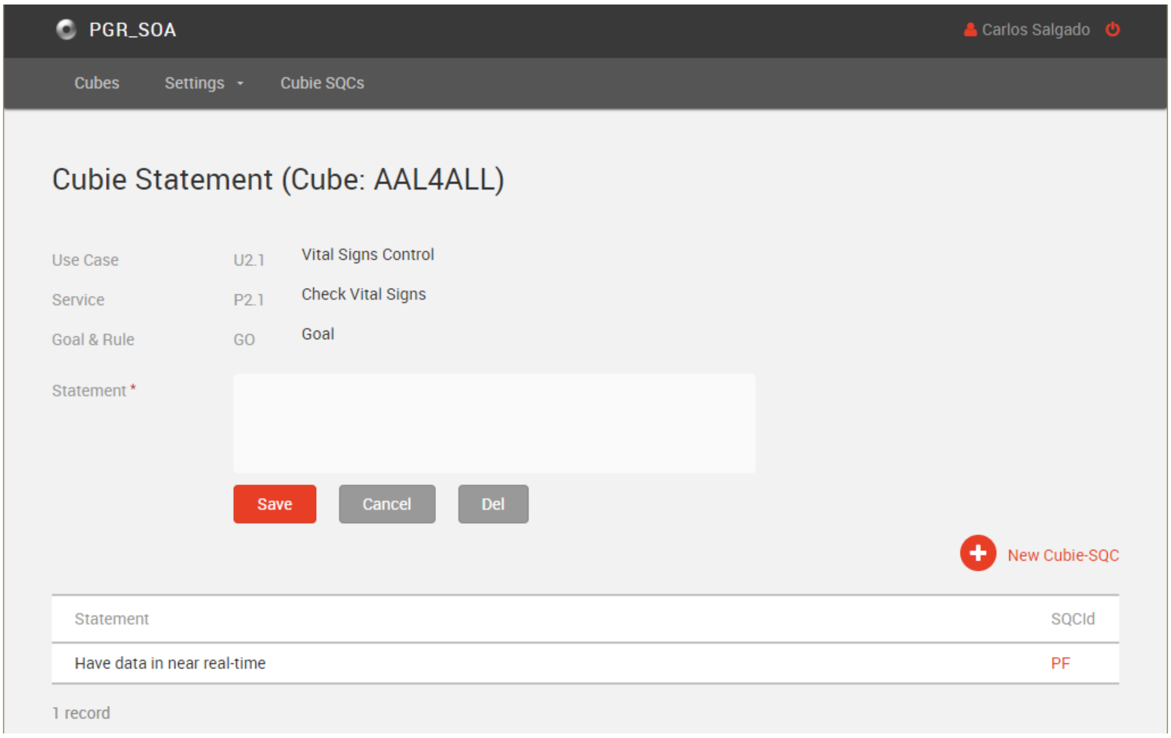
[PGR-SOA] Detail of a PGR-SOA Cube:



[PGR-SOA] Use-case Perspective of a PGR-SOA Cube:



[PGR-SOA] Detail of a PGR-SOA Cubie Statement and associated SQC Statements:



Disclaimer: As the related projects' documents contain material, which is the copyright of certain consortium parties, it may not be reproduced or copied without permission, so hindering further details in these appendices.